

Spring Cloud

微服务实战

翟永超 著

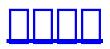




Spring Clo	ud[][][

Spring Cloud
00000Java000000000000000000000000000000
Spring Cloud—2017.5
ISBN 978-7-121-31301-1
I.①S II.①[] III.①[][][—[][][] IV.①TP368.5
CIP2017071305_
□□□787×980 1/16
□□□27.5

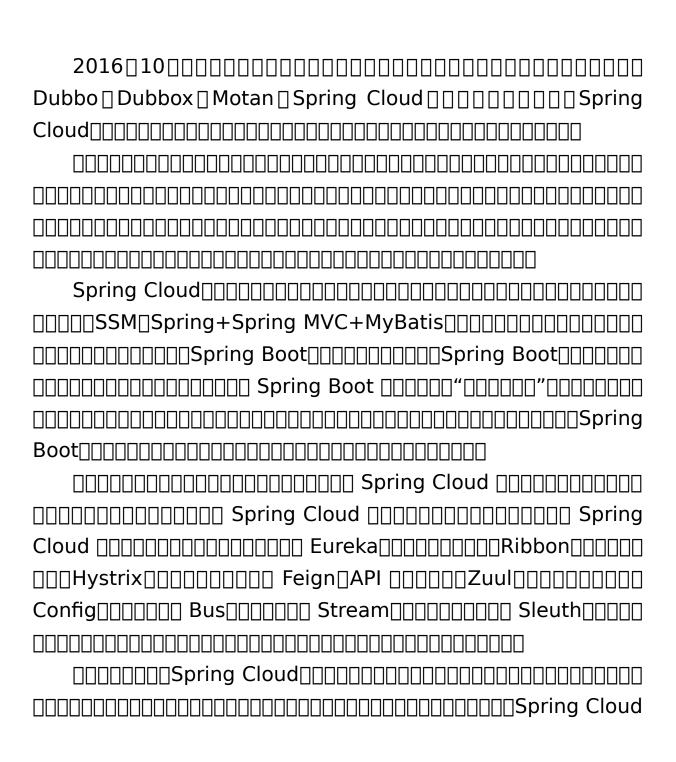
010_88254888 <u></u> 88258888	
dbqq@phei.com.cn[]	
0010-51260888-819 faq@phei.com.cn	



Java2003
Struts/JSP/Hibernate/MySQL $\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \$
$\ \ \square \ \text{http://www.fujitsu.com/cn/products/software/applications/s}$
pif/[[
J2EESpringEMC
$Spring \verb EMC VMware,VMware Spring 2012 $
<pre>Domain Data/XD Description Data/XD Description Description</pre> <pre>Data/XD Description Description Description</pre>
$Data\ Pipeline \verb $
2015
Cloud-Native Application Architectures \cite{res} DevOps \cite{res}
NetflixOSS
0002015090000000000000000000000000000000
000Spring000000000000000000000000000000000000
Spring Boot
Cloud
CloudCI/CD
Spring Cloud
Spring Cloud

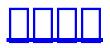
Spring Cloud
Spring Cloud Config
000000000000000000000000000000000000000
Spring Cloud
Spring CloudSpring Cloud
Spring
Cloud
Spring Cloud
DaoCloud 🗆 🗆
2017[]3[]





Spring Cloud

2017.03.27



DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
really knows how to do it, everyone thinks everyone else is
doing it,so everyone claims they are doing it."
EDAS Dubbo
Docker
Spring Cloud



Spring Cloud
00000000000000000000000000000000000000
Cloud
Spring Cloud
Spring Cloud
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
Cloud

Spring Cloud
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
Cloud
□□□xujin.org□
Spring Cloud
Spring Cloud
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD



___Spring Cloud______ _____Spring Cloud NOTE THE Spring Cloud _____Spring Cloud _____Spring Cloud TOTO TO Spring Cloud On

00000000000000000000000000000000000000
Cloud
OOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO
Spring Cloud
Spring Cloud
Spring Cloud
www.broadview.com.cn
000000000000000000000000000000000000000
http://www.broadview.com.cn/31301



□□□□Spring Cloud	
Spring Cloud□□	
□2□ □□□□□Spring Boot	
□□RESTful API	

```
□□actuator
   □3□ □□□□□Spring Cloud Eureka
 Netflix Eureka
   <u>Eureka</u>∏
   <u>RestTemplate</u>□□
   <u>GET</u>□□
   <u>POST</u>□□
   <u>PUT</u>∏
   DELETE
```

```
\underline{Camden} \underline{\sqcap} \underline{\sqcap} \underline{RibbonClient} \underline{\sqcap} \underline{\sqcap} \underline{\sqcap}
      <u>□Eureka</u>□□
   <u>collapser</u>∏
      <u>threadPool</u>□□
   <u>Hystrix</u>∏∏
   <u>Turbine</u>□□□□
```

```
□6□ □□□□□□□Spring Cloud Feign
 Ribbon□□
  <u>Hystrix</u>∏
  □□Hystrix
  □7□ API□□□□□Spring Cloud Zuul
```

```
Cookie□□□□
   <u>Hystrix</u> □ Ribbon □ □
 □8□ □□□□□□□□Spring Cloud Config
 <u>Git</u>□□□
   SVN
```

```
<u>URIDDDDD</u>
    <u>□9□ □□□□□Spring Cloud Bus</u>
  <u>RabbitMQ</u>□□□□□
    □□Spring Cloud Bus
    RabbitMQ□□
  Kafka∏∏∏∏∏
    <u>Kafka</u>∏
    □□Spring Cloud Bus
    Kafka∏∏
  □10□ □□□□□□□□Spring Cloud Stream
```

```
<u>RabbitMQ</u> <u>Kafka</u> □□
 □11□ □□□□□□□Spring Cloud Sleuth
```

```
□Logstash□□
□Zipkin□□
HTTP□□
□□□□□□
□□□□
□□□□
API□□
□□A Starter POMs
```



Web	
000000000000000000000000000000000000000	

Martin
Fowler Microservices DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
00000000000000000000000000000000000000
0000000000000 PC 0000000000000000000000
00000000000000000000000000000000000000
00000000000000CPU0000000000 PC 000000000

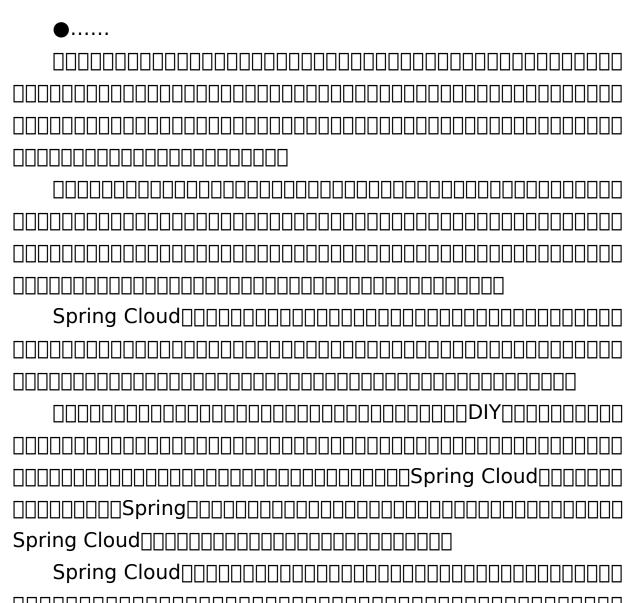
00000000"00"000000000000000000000000000
●□□□□□□HTTP□RESTful API□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
●□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
protobuf
00000000"0000000"000000000000000000000
——Martin Fowler

MongoDB
□Redis□□□
ullet
ullet

Spring Cloud

000000000000000000000000000000000000000

- ●□□□□□□□□□□Elastic-Job□LinkedIn□Azkaban□Spring Cloud
 □Task□□
- □ □ □ □ □ □ □ Hydra □ Spring Cloud □ Sleuth □ Twitter □ Zipkin □ □



Spring Cloud

Spring Cloud
Spring Cloud
●Spring Cloud Config□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
●Spring Cloud Netflix□□□□□□□□Netflix OSS□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
■Eureka
■Hystrix
■Ribbon□□□□□□□□□□□□□
■Feign□□□Ribbon□Hystrix□□□□□□□□□□□
■Zuul00000000000000000000
■Archaius□□□□□□□
●Spring Cloud Bus□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
●Spring Cloud Cluster□□□ZooKeeper□Redis□Hazelcast□
●Spring Cloud Cloudfoundry□□Pivotal Cloudfoundry□□□□
●Spring Cloud Consul□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
●Spring Cloud Stream□□□Redis□Rabbit□□Kafka□□□□□□□
●Spring Cloud AWS□□□□□□□Amazon Web Service□□□□
●Spring Cloud Security□□□□□□□□Zuul□□□□□OAuth2□□□□

 Spring Cloud Sleuth:Spring Cloud Cloud

	An	gel.SR6[Brixton.SR5[] SR6[
SR5		
□□□□□Spri	ng Cloud	bug
"service	releases" 🛮 🖺 🗎 🖺	SRX [] [] [] [X [] [] [] [] [] [
Brixton.SR5□□B	rixton <u> </u> 5 Relea	se[[[
		000000000000000000000000000000000000000
Componer	nt	Angel.SR6
Brixton.SR5	Camden.	M1 Camden.BUILD
SNAPSHOT		
spring-clou	d-aws	1.0.4.RELEASE
1.1.1.RELEASE	1.1.1.RELEASE	1.1.2.BUILD-SNAPSHOT
spring-clou	d-bus	1.0.3.RELEASE
1.1.1.RELEASE	1.2.0.M1	1.2.0.BUILD-SNAPSHOT
spring-clou	d-cli	1.0.6.RELEASE
1.1.5.RELEASE	1.2.0.M1	1.2.0.BUILD-SNAPSHOT
spring-clou	d-commons	1.0.5.RELEASE
1.1.1.RELEASE	1.1.1.RELEASE	1.1.2.BUILD-SNAPSHOT
spring-clou	d-contract	
1.0.0.M2	1.0.0.BUILD-SNA	APSHOT
spring-clou	d-config	1.0.4.RELEASE
1.1.3.RELEASE	1.2.0.M1	1.2.0.BUILD-SNAPSHOT
spring-clou	d-netflix	1.0.7.RELEASE
1.1.5.RELEASE	1.2.0.M1	1.2.0.BUILD-SNAPSHOT
spring-clou	d-security	1.0.3.RELEASE
1.1.2.RELEASE	1.1.2.RELEASE	1.1.3.BUILD-SNAPSHOT

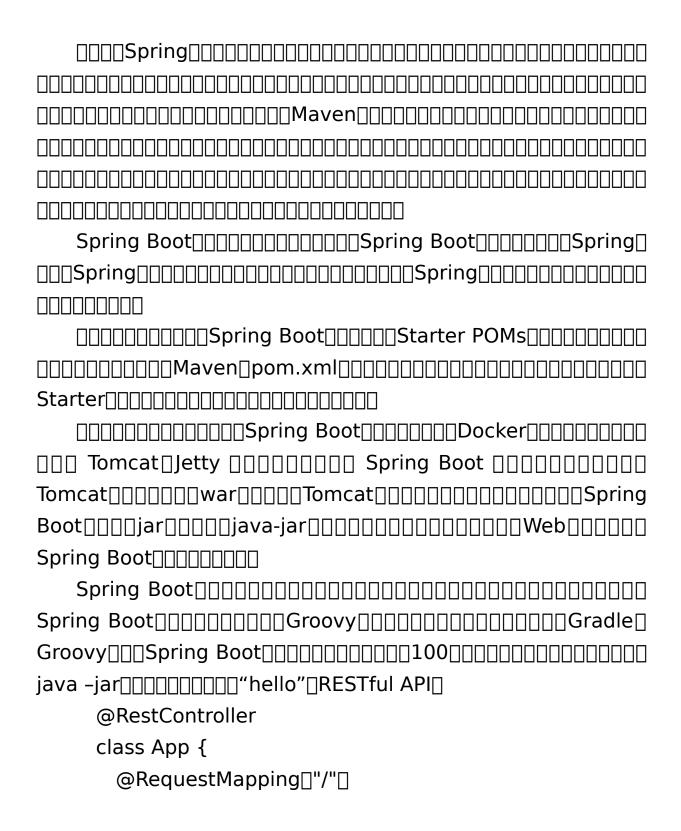
	spring-cloud	l-starters	1.0.6.RELEA	SE	
	spring-cloud	l-cloudfoundry			
1.0	.0.RELEASE	1.0.0.RELEAS	E 1.0.1.BUIL	D-SNAPSHO	T
	spring-cloud	l-cluster]	L.O.1.RELEAS	SE
	spring-cloud	l-consul		1.0.2.RELE	ASE
1.1	.0.M1	L.1.0.BUILD-SN	IAPSHOT		
	spring-cloud	l-sleuth		1.0.6.RELE	ASE
1.0	.6.RELEASE	1.0.7.BUILD-S	NAPSHOT		
	spring-cloud	l-stream		1.0.2.RELE	ASE
Bro	oklyn.M1	Brooklyn.BUIL	D-SNAPSHOT	-	
	spring-cloud	l-zookeeper		1.0.2.RELE	ASE
1.0	.2.RELEASE	1.0.3.BUILD-S	NAPSHOT		
	spring-boot	1.2	2.8.RELEASE	1.3.7.RELE	ASE
1.4	.0.RELEASE	1.4.0.RELEAS	E		
	spring-cloud	l-task		1.0.2.RELE	ASE
1.0	.2.RELEASE	1.0.3.BUILD-S	NAPSHOT		
	Ar	ngel]Brixt@	on[]Camden[
]][][]Brixton [Camden □□□	
	Brixton		g Cloud	Release	
]][][]Brixton.S	SR5Spri	ng Boot 1.3.	.7000
		□ □ Brixton □ □			
Bri	xton.SR7∏	Brixt	:on[][][][][] □Spring B o	oot∏□
][[[] 1.3. x[][1.4. x			pring
1.4	. x	Spring Cloud	Camd	en[[[[[[Cai	mden
] Release	0000000000	C a	mden.SR3	
Car	mden[][][[][Brixton[[

Cloud[][][][][]

2 | Spring Boot

One of the contraction of the co
Spring BootSpring Boot
□□□□Spring Cloud□□□□□□□
One Spring Boot One
Cloud
OSpring Boot
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
Spring Cloud
●□□□□Spring Boot□□
●□□□□RESTful API□□
●□□□□□□Spring Boot□□□□
●□□□Spring Boot□□□□□□
●Spring Boot□□□□□□□
Spring Boot

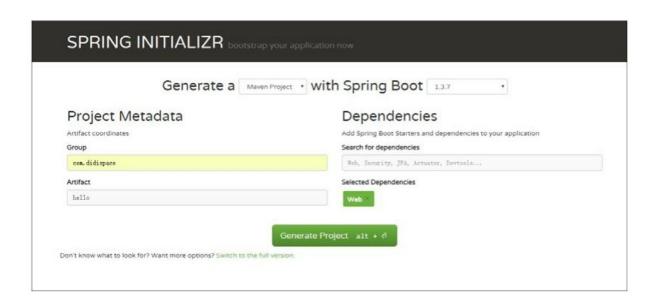




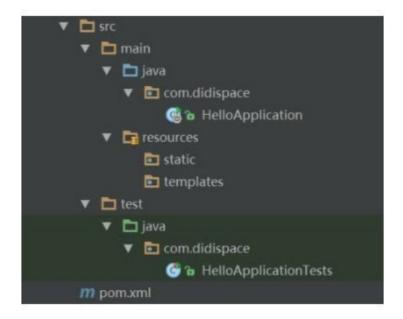
```
String home □ [
    "hello"
   }
  }
 ____Spring Boot______
Spring Boot
 _____Spring Boot______
RESTful API
●Java 7□□□□□
 ●Spring Framework 4.2.7□□□□□
 □□□□□□Java 1.8.0 73□Spring Boot 1.3.7□□□□□
 □□Maven□□
 2. | http://start.spring.io/ Maven Gradle
□□Spring Boot□□□□□□
 3. ☐☐☐☐☐ Maven Project☐Spring Boot ☐☐☐☐1.3.7☐☐☐ Group
```

__RESTful API____Web___

- 6. | File--> New--> Project from Existing Sources...



- - 9.____JDK___Java SDK____Java 7_____ _____



- _____ pom.xml _____ Spring Boot

<? xml version="1.0" encoding="UTF-8"? >

```
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0h
ttp://maven.apache.org/xsd/maven-4.0.0.xsd">
     <modelVersion>4.0.0</modelVersion>
     <groupId>com.didispace
     <artifactId>spring-boot-hello</artifactId>
     <version>0.0.1-SNAPSHOT
     <packaging>jar</packaging>
     <name>spring-boot-hello</name>
     <description>Demo
                                               Spring
                            project
                                       for
 Boot</description>
     <parent>
       <groupId>org.springframework.boot</groupId>
       <artifactId>spring-boot-starter-parent</artifactId>
       <version>1.3.7.RELEASE
       <relativePath/> <!--lookup parent from repository-
   ->
     </parent>
     properties>
       project.build.sourceEncoding>UTF-
   8</project.build.sourceEncoding>
       project.reporting.outputEncoding>UTF-
   8</project.reporting.outputEncoding>
       <java.version>1.8</java.version>
     </properties>
     <dependencies>
       <dependency>
         <groupId>org.springframework.boot</groupId>
```

```
</dependency>
      <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
      </dependency>
    </dependencies>
     <bul>d
      <plugins>
       <plugin>
        <groupId>org.springframework.boot</groupId</pre>
      >
        <artifactId>spring-boot-maven-
      plugin</artifactId>
       </plugin>
      </plugins>
    </build>
   </project>
  ____groupId_artifactId________
Description:
Spring Boot
_____Tomcat
____ Spring Boot ______
application.properties□□□□□
```

<artifactId>spring-boot-starter-web</artifactId>

| dependencies |
|---|
| ● spring-boot-starter-web 🛮 🖺 🖺 Web 🗎 🗎 🗎 🗎 🖺 Tomcat 🗈 |
| Spring MVC |
| ● spring-boot-starter-test [] [] [] [] [] JUnit [Hamcrest [|
| Mockito□ |
| <pre>[][][][]web[]test[][][]Spring Boot[][][][]Starter POMs[]</pre> |
| Starter POMsSpringSpring |
| Web |
| spring-boot-starter-web |
| spring-boot-starter-jdbc [] [] [] [] spring-boot-starter-data- |
| jpa Spring Boot |
| pom.xmlStarter POMs |
| |
| Spring Boot Starter POMs Spring-boot-starter-* |
| Boot |
| Starter POMsSpring |
| Cloud |
| Spring |
| Boot 1.3.7 |
| |
| buildSpring Boot_Maven |
| jarmvn |
| spring-boot:run∏∏∏∏∏∏∏Spring Boot∏∏ |

```
□Spring Boot□□□□□RESTful API□□□□□□Spring MVC□□□□□□□□
@RestController
             public class HelloController {
                  }
                  @RequestMapping□"/hello"□
                  public String index□□{
                      return "Hello World":
              }
         ●□□□□□□□□□□□□http://localhost:8080/hello□□□□□□□□□□
□□□□Hello World□
         □□Spring Boot□□
         □□Spring Boot□□□□□□□□□
         ullet 
spring-boot (org.springframework.boot:spring-boot-maven-plugin:1.3.7.RELEASE)
              👺 spring-boot:help
              豫 spring-boot:repackage
              👺 spring-boot:run
              👺 spring-boot:start
              👺 spring-boot:stop
         xxx.jar∏∏∏∏∏
```

```
com.didispace.HelloApplicationTests
On the local desired from the local desired f
           @RunWith∏SpringJUnit4ClassRunner.class∏
                 @SpringApplicationConfiguration
      □classes=HelloApplication.class□
                 @WebAppConfiguration
                 public class HelloApplicationTests {
                       private MockMvc mvc;
                       @Before
                       public void setUp∏∏throws Exception {
                             mvc=MockMvcBuilders.standaloneSetup
                                                                                                                                                                       new
                 HelloController□□□.build□□;
                       }
                       @Test
                       public void hello□□throws Exception {
                             mvc.perform
                                                                                             MockMvcRequestBuilders.get
                 □"/hello"□.accept□MediaType.APPLICATION JSON□□
                                   .andExpect[]status[][].isOk[][][]
                                   .andExpect □ content □ □ .string □ equalTo □ "Hello
                       World"□□□;
                        }
```

| } |
|---|
| |
| @RunWith Spring JUnit 4 Class Runner.class Spring |
| JUnit4 🗆 🗆 🗆 |
| @SpringApplicationConfiguration |
| <pre>□classes=HelloApplication.class□□□□Spring Boot□□□□□</pre> |
| ● @WebAppConfiguration □ □ □ Web □ □ □ □ □ □ □ □ □ |
| ServletContext[] |
| ●MockMvc□□□□□□□□□Controller□□□□□□□□□@Test□□□hello |
| perform accept andExpect |
| |
| ●@Before:JUnit□□□□□□@Test□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| HelloController 🗆 🗆 🗆 |
| |
| import static org.hamcrest.Matchers.equalTo; |
| import static |
| org.springframework.test.web.servlet.result.MockMvcResul |
| tMatchers.content; |
| import static |
| org.springframework.test.web.servlet.result.MockMvcResul |
| tMatchers.status; |
| |
| |
| Boot |
| DDDDDDDC"Hello World"DRESTful |
| Spring |

| Cloud |
|---|
| |
| |
| |
| Spring Boot [] [] [] [] [] [] [] [] [] [|
| Spring Boot |
| src/main/resources/application.properties Spring Boot Spring Boot |
| Starter |

| Web |
|--|
| $application.properties \verb \verb server.port = 8888 \verb \verb 8888 \verb $ |
| $\verb $ |
| |
| Spring Boot |
| |
| YAML/jæməl/camel |
| $\verb $ |
| $ \ \square RFC 2822 \\ \ \square \square \square \square \square \square \square \square Clark Evans \\ \ \square 2001 \\ \ \square Ingy d\"{ot} \\$ |
| Net Oren Ben-Kiki |
| YAML YAML Ain't a Markup Language YAML |
| |
| Language |
| |
| |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| ——0000 |
| YAML |
| |
| environments: |
| dev: |
| url:http://dev.bar.com |
| name: Developer Setup |
| prod: |

```
url:http://foo.bar.com
    name: My Cool App
 Documentation
  environments.dev.url=http://dev.bar.com
  environments.dev.name=Developer Setup
  environments.prod.url=http://foo.bar.com
  environments.prod.name=My Cool App
 server:
   port: 8881
  spring:
   profiles: test
  server:
   port: 8882
  spring:
   profiles: prod
  server:
   port: 8883
 YAML
```

```
____Spring Boot_____Starter_______
On the state of th
                    book.name=SpringCloudInAction
                    book.author=ZhaiYongchao
             nnnnnnnn@Valuennnnnnnnnnnnn
                    @Component
                    public class Book {
                           @Value[]"${book.name}"[]
                           private String name;
                           @Value∏"${book.author}"∏
                           private String author;
                          //∏getter∏setter
                     }
             @Value
             ●□□□□□□□□□PlaceHolder□□□□□□${...}□□□□□□PlaceHolder□
             ●□□□□□□SpEL□□□□Spring Expression Language□□□□□#
 \{...\}
                                                                                               book.name=SpringCloud
                    book.author=ZhaiYongchao
```

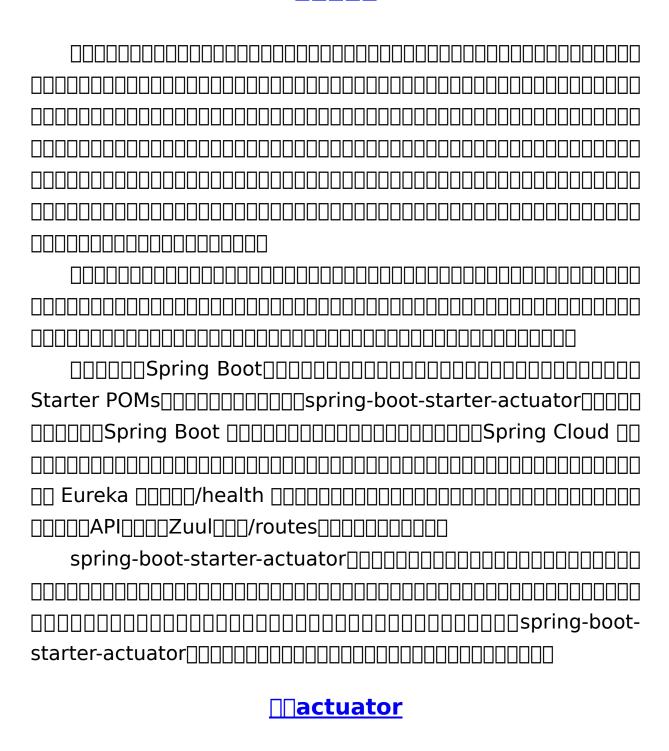
| book.desc@@@@@@book.name@book.author@@@@@ |
|--|
| □□□□ZhaiYongchao is writing□SpringCloud□□ |
| |
| |
| com.didispace.blog.value=\${random.value} # int |
| com.didispace.blog.number=\${random.int} # |
| com.didispace.blog.bignumber=\${random.long} # 10000000 |
| com.didispace.blog.test1= ${\text{random.int}}10$ } # 10 |
| com.didispace.blog.test2=\${random.int[10,20]} |
| |
| |

| Spring Boot |
|--|
| application.properties <code>[][][][][][][][][][][][][][][][][][][]</code> |
| $server.port = 8888 \ \square \ application.properties \ \square \ $ |
| server.port=8888[] |
| Spring Boot |
| |
| Spring Maven Profile Control Spring Boot Control Profile Control Control Profile Pro |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| |
| |
| |
| |
| |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Spring Boot [] [] [] [] [] [] [] application- |
| {profile}.properties |
| ●application-dev.properties□□□□□□ |
| ●application-test.properties□□□□□□ |
| ●application-prod.properties□□□□□□ |
| |
| spring.profiles.active |
| $spring.profiles.active = test \verb application-test.properties $ |
| |

| ● □ □ □ □ □ □ □ □ □ □ □ application-dev.properties □ |
|--|
| application-test.properties[application-prod.properties[|
| ●□□□□□□□□□□□server.port□□□□□□dev□□□□□1111,test□□ |
| 2222,prod3333_ |
| <pre>application.properties[][]spring.profiles.active=dev[][]</pre> |
| dev |
| |
| ● java-jar xxx.jar |
| <pre>□dev□□</pre> |
| ●□□java-jar xxx.jarspring.profiles.active=test□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| test |
| ●□□java-jar xxx.jarspring.profiles.active=prod□□□□□□□□□ |
| 3333prod |
| |
| ● □ application.properties □ □ □ □ □ □ □ □ □ |
| spring.profiles.active=dev |
| ■□application-{profile}.properties□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| |
| |
| Spring Boot |
| spring.profiles.active |
| QA |
| |
| |

| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
|--|
| Config DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Spring Boot |
| Spring Boot |
| 1.00000000 |
| 2.SPRING_APPLICATION_JSON |
| SPRING_APPLICATION_JSON DDJSONDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| 3.java:comp/env[[]]NDI[[[] |
| 4.Java |
| 5.00000000 |
| 6. random.* |
| 7 jar{profile}application- |
| {profile}.properties[][YAML]][][][] |
| 8 jar{profile}application- |
| {profile}.properties[][YAML][][][][] |
| 9jarapplication.properties_YAML |
| 10.00000jar0000application.properties0YAML00000 |
| 11. @Configuration Description Property Source Description |
| |
| 12.000000 SpringApplication.setDefaultProperties |
| |
| |
| 79jar |
| jarjarjar |
| |
| |





```
____ Spring Boot _____ pom.xml _
<dependency>
                          <groupId>org.springframework.boot</groupId>
                          <artifactId>spring-boot-starter-actuator</artifactId>
                    </dependency>
             : Mapped "{[/info || /info.json],methods=[GET],produces=[application/
: Mapped "{[/env/{name:.*}],methods=[GET],produces=[application/json]
                                                                            : Mapped "{[/env || /env.json],methods=[GET],produces=[application/js
                                                                           : Mapped "{[/dump || /dump.json],methods=[GET],produces=[application/
: Mapped "{[/autoconfig || /autoconfig.json],methods=[GET],produces=[
      o.s.b.a.e.mvc.EndpointHandlerMapping
                                                                           : Mapped "{[/health || /health.json],produces=[application/json]}
     o.s.b.a.e.mvc.EndpointHandlerMapping
                                                                           : Mapped "{[/mappings || /mappings.json],methods=[GET],produces=[appli
: Mapped "{[/beans || /beans.json],methods=[GET],produces=[application
    o.s.b.a.e.mvc.EndpointHandlerMapping
                                                                           : Mapped "{[/configprops || /configprops.json],methods=[GET],produces
     o.s.b.a.e.mvc.EndpointHandlerMapping
     o.s.b.a.e.mvc.EndpointHandlerMapping
             _____spring-boot-
starter-actuator
nnnnnnnnnnnnnnhhealthnnnnnnnnnnnn
                    {
                          "status": "UP",
                          "diskSpace": {
                                "status": "UP",
                                "total": 491270434816,
                                "free": 383870214144,
                                "threshold": 10485760
                          }
             NOTIONAL TO BE STORED TO THE PROPERTY OF THE P
```

| ●□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
|---|
| |
| ●□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| |
| |
| |
| |
| Spring Boot Company Company |
| |
| |
| 00000000000000000000000000000000000000 |
| |
| ●/autoconfig |
| |
| |
| ■positiveMatches □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ |
| ■negativeMatches□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |

```
{
             "bean":
         "org.springframework.boot.autoconfigure.web.
   DispatcherServletAutoConfiguration$DispatcherServlet
Configuration",
             "scope": "singleton",
             "type":
         "org.springframework.boot.autoconfigure.web.
   DispatcherServletAutoConfiguration$DispatcherServlet
Configuration$$EnhancerBySpring CGLIB$$3440282b",
             "resource": "null",
             "dependencies":[
               "serverProperties",
               "spring.mvc.CONFIGURATION_PROPERTIES"
               "multipartConfigElement"
           },
           {
             "bean": "dispatcherServlet",
             "scope": "singleton",
             "type":
         "org.springframework.web.servlet.DispatcherSer
         vlet",
             "resource": "class path resource
   [org/springframework/boot/autoconfigure/web/Dispatch
erServletAutoConfiguration$Disp
```

```
atcherServletConfiguration.class]",
         "dependencies":[]
        }
       ]
      }
  ■bean:Bean□□□□
  ■type:Bean | Java | | | |
  ■resource:class
  ■dependencies □ □ □ Bean □ □ □
  _____properties
endpoints.configprops.enabled=false
   {
    "configurationPropertiesReportEndpoint": {
      "prefix": "endpoints.configprops",
      "properties": {
       "id": "configprops",
       "sensitive": true,
       "enabled": true
      }
    },
```

```
nnnnnnnnnnnnnn passwordnsecretnkey nnnnnnnnnnnn
{
     "profiles":[
      "dev"
     ],
     "server.ports": {
      "local.server.port": 8881
     },
     "servletContextInitParams": {
     },
     "systemProperties": {
      "idea.version": "2016.1.3",
      "java.runtime.name": "Java □ TM □ SE Runtime
   Environment".
      "sun.boot.library.path":
                              "C:\\Program
   Files\\Java\\jdk1.8.0 91\\jre\\bin",
      "java.vm.version": "25.91-b15",
      "java.vm.vendor": "Oracle Corporation",
      . . .
     },
     "systemEnvironment": {
```

```
"configsetroot": "C:\\WINDOWS\\ConfigSetRoot",
        "RABBITMQ BASE": "E:\\tools\\rabbitmq",
      },
      "applicationConfig:[classpath:/application-
   dev.properties]": {
        "server.port": "8881"
      },
      "applicationConfig:
   [classpath:/application.properties]": {
        "server.port": "8885",
        "spring.profiles.active": "dev",
        "info.app.name": "spring-boot-hello",
        "info.app.version": "v1.0.0",
        "spring.application.name": "hello"
      }
    }
   bean_____method______
      {
        "/webjars/**": {
         "bean": "resourceHandlerMapping"
       },
        "/**": {
         "bean": "resourceHandlerMapping"
        },
```

```
"/**/favicon.ico": {
        "bean": "faviconHandlerMapping"
       },
       "{[/hello]}": {
        "bean": "requestMappingHandlerMapping",
        "method":
                      "public
                                 java.lang.String
      com.didispace.web.HelloController.index□□"
       },
       "{[/mappings
                     Ш
                         /mappings.json],methods=
    [GET],produces=[application/json]}": {
        "bean": "endpointHandlerMapping",
        "method": "public java.lang.Object
    org.springframework.boot.actuate.endpoint.mvc.Endpo
 intMvcAdapter.invoke∏∏"
       },
      }
   info.app.name=spring-boot-hello
    info.app.version=v1.0.0
   {
      "app": {
       "name": "spring-boot-hello",
       "version": "v1.0.0"
```

```
}
                 ON THE THE TERM OF THE PROPERTY OF THE PROPERT
{
                                 "mem": 541305,
                                  "mem.free": 317864,
                                  "processors": 8,
                                  "instance.uptime": 33376471,
                                  "uptime": 33385352,
                                  "systemload.average":-1,
                                  "heap.committed": 476672,
                                  "heap.init": 262144,
                                  "heap.used": 158807,
                                  "heap": 3701248,
                                  "nonheap.committed": 65856,
                                  "nonheap.init": 2496,
                                  "nonheap.used": 64633,
                                  "nonheap": 0,
                                  "threads.peak": 22,
                                  "threads.daemon": 20,
```

```
"threads.totalStarted": 26.
      "threads": 22.
      "classes": 7669,
      "classes.loaded": 7669,
      "classes.unloaded": 0,
      "gc.ps scavenge.count": 7,
      "gc.ps_scavenge.time": 118,
      "gc.ps_marksweep.count": 2,
      "gc.ps marksweep.time": 234,
      "httpsessions.max":-1,
      "httpsessions.active": 0,
      "gauge.response.beans": 55,
      "gauge.response.env": 10,
      "gauge.response.hello": 5,
      "gauge.response.metrics": 4,
      "gauge.response.configprops": 153,
      "gauge.response.star-star": 5,
      "counter.status.200.beans": 1,
      "counter.status.200.metrics": 3,
      "counter.status.200.configprops": 1,
      "counter.status.404.star-star": 2,
      "counter.status.200.hello": 11.
      "counter.status.200.env": 1
   instance.uptime
```

| ■mem.*□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
|---|
| java.lang.Runtime□ |
| ■ heap.* |
| java.lang.management.MemoryMXBean |
| getHeapMemoryUsage |
| java.lang.management.MemoryUsage□ |
| ■ nonheap.* |
| java.lang.management.MemoryMXBean |
| getNonHeapMemoryUsage 🛮 🗎 🖺 🗎 |
| java.lang.management.MemoryUsage□ |
| ■ threads.* |
| peak |
| ■ classes.* |
| java.lang.management.ClassLoadingMXBean□ |
| ■gc.* |
| gc.ps_scavenge.time |
| gc.ps_marksweep.count [|
| gc.ps_marksweep.time |
| java.lang.management.GarbageCollectorMXBean□ |
| ■ httpsessions.*:Tomcat □ □ □ □ □ □ □ □ □ □ □ □ □ □ |
| httpsessions.max |
| □Tomcat□□□□□□□□□□□ |
| ■gauge.*:HTTP□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| gauge.response.hello: 5000000hello0000005000 |
| ■counter.*:HTTP |
| DDDDCounter.status.200.hello: 11DDDDhelloDDD200DDDD |
| |

| gauge.* counter.* |
|--|
| |
| |
| $org.spring framework.boot.actuate.metrics. Counter Service \square$ |
| $org.spring framework.boot.actuate.metrics. Gauge Service \ \ \square \ \square$ |
| 00000000000000000000000000000000000000 |
| @RestController |
| <pre>public class HelloController {</pre> |
| @Autowired |
| private CounterService counterService; |
| @RequestMapping[]"/hello"[] |
| public String greet□□{ |
| counterService.increment |
| <pre>"didispace.hello.count"[];</pre> |
| return ""; |
| } |
| } |
| /metrics |
| |
| /metrics/{name} |
| /metrics/mem.free |
| ●/health□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| spring-boot-starter-actuator |
| |

| 检测器 | 功能 |
|---------------------------|-----------------------|
| DiskSpaceHealthIndicator | 低磁盘空间检测 |
| DataSourceHealthIndicator | 检测 DataSource 的连接是否可用 |
| MongoHealthIndicator | 检测 Mongo 数据库是否可用 |
| RabbitHealthIndicator | 检测 Rabbit 服务器是否可用 |
| RedisHealthIndicator | 检测 Redis 服务器是否可用 |
| SolrHealthIndicator | 检测 Solr 服务器是否可用 |

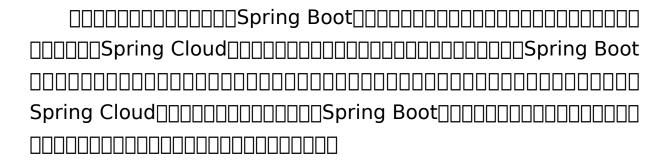
```
_____ Boot _____
org.springframework.boot.actuate.health.HealthIndicator
@Component
             RocketMOHealthIndicator
   public class
                              implements
 HealthIndicator {
    @Override
    public Health health □ {
      int errorCode=check∏∏;
      if∏errorCode !=0∏{
       return Health.down □ □ .withDetail □ "Error
    Code",errorCode∏.build∏;
      return Health.up□□.build□□;
    }
    private int check□□{
      }
```

```
"timestamp": 1482570022463,
             "info": {
                "method": "GET",
                "path": "/metrics/mem",
                "headers": {
                   "request": {
                      "host": "localhost:8881",
                       "connection": "keep-alive",
                       "cache-control": "no-cache",
                       "user-agent": "Mozilla/5.0 (Windows NT 10.0; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36",
                       "postman-token": "9817ea4d-ad9d-b2fc-7685-9dff1a1bc193",
                       "accept": "*/*",
                       "accept-encoding": "gzip, deflate, sdch",
                      "accept-language": "zh-CN, zh; q=0.8"
                   },
                   "response": {
                       "X-Application-Context": "hello:dev:8881",
                       "Content-Type": "application/json; charset=UTF-8",
                       "Transfer-Encoding": "chunked",
                       "Date": "Sat, 24 Dec 2016 09:00:22 GMT",
                       "status": "200"
          },
```

endpoints.shutdown.enabled=true





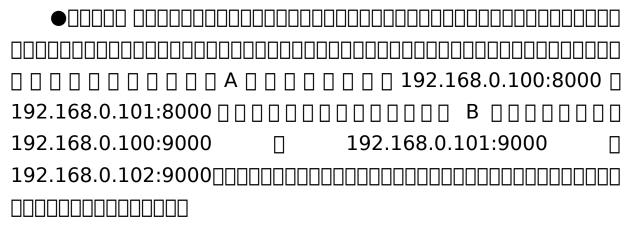


3 □ □ □ □ Spring Cloud Eureka

- ullet
- ullet
- ●Eureka[][][]
- ●Eureka[][][][]
- ●Eureka[][]



| 00000000000000000000B00000000000000000 |
|--|
| |
| |
| |



| 服务名 | 位置 |
|------|--|
| 服务A | 192.168.0.100:8000、192.168.0.101:8000 |
| 服务 B | 192.168.0.100:9000、192.168.0.101:9000、192.168.0.102:9000 |

| | .00000000000000000000000000000000000000 | |
|--|---|--|
| | | |

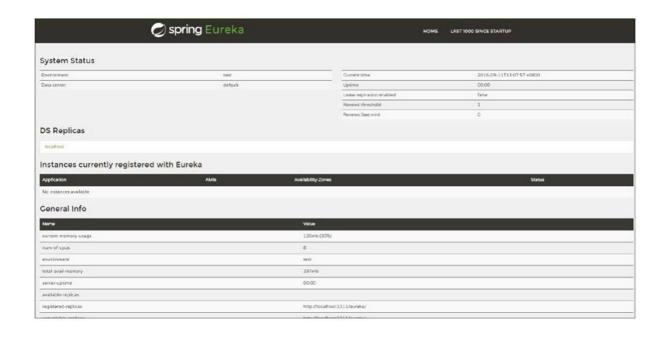
Netflix Eureka

| Spring Cloud Eureka Netflix Eureka |
|--|
| |
| JavaEurekaJVMEureka |
| RESTful API Java Eureka |
| 00000000000000000000000000000000000000 |
| EurekaNET |
| Steeltoe Node.js eureka-js-client |
| Eureka |
| 00000000000000000000000000000000000000 |
| 00000000 Eureka 000000000000000000000000000000000000 |
| 00000000000000000000000000000000000000 |
| Netflix DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| Eureka |
| 00000000000000000000000000000000000000 |
| |
| Eureka |
| |
| |

```
<artifactId>spring-boot-starter-parent</artifactId>
      <version>1.3.7.RELEASE
      <relativePath/>
    </parent>
    <dependencies>
      <dependency>
       <groupId>org.springframework.cloud</groupId>
       <artifactId>spring-cloud-starter-eureka-
    server</artifactId>
      </dependency>
    </dependencies>
    <dependencyManagement>
      <dependencies>
       <dependency>
         <groupId>org.springframework.cloud</groupId>
         <artifactId>spring-cloud-
      dependencies</artifactId>
         <version>Brixton.SR5</version>
         <type>pom</type>
         <scope>import</scope>
       </dependency>
      </dependencies>
    </dependencyManagement>
   _____Spring Boot_______
    @EnableEurekaServer
```

<groupId>org.springframework.boot</groupId>

```
@SpringBootApplication
   public class Application {
    public static void main [] String[] args [] {
                      SpringApplicationBuilder
      new
   □Application.class□.web□true□.run□args□;
    }
   }
  server.port=1111
   eureka.instance.hostname=localhost
   eureka.client.register-with-eureka=false
   eureka.client.fetch-registry=false
   eureka.client.serviceUrl.defaultZone=http://${eureka.i
 nstance.hostname}:${serve r.port}/eureka/
  server.port
  false
  ____false_
  _____http://localhost:1111/_____
Eureka [ ] [ ] [ ] Instances currently registered with Eureka [ ] [ ]
```





</dependency>

<dependency>

```
<groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-
    eureka</artifactId>
      </dependency>
     </dependencies>
     <dependencyManagement>
      <dependencies>
        <dependency>
          <groupId>org.springframework.cloud
          <artifactId>spring-cloud-
      dependencies</artifactId>
          <version>Brixton.SR5</version>
          <type>pom</type>
          <scope>import</scope>
        </dependency>
      </dependencies>
     </dependencyManagement>
   @RestController
    public class HelloController {
                             logger=Logger.getLogger
      private
               final
                     Logger
   □getClass□□□;
      @Autowired
      private DiscoveryClient client;
      @RequestMapping
   □value="/hello",method=RequestMethod.GET□
```

```
public String index□□{
                   ServiceInstance
           instance=client.getLocalServiceInstance∏;
                   logger.info | "/hello,host:"+instance.getHost | |
           +",service id:"+instance.getServiceId[[]];
                   return "Hello World":
               }
           }
       Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Descripti
@EnableDiscoveryClient
           @SpringBootApplication
           public class HelloApplication {
               public static void main [] String[] args [] {
                   SpringApplication.run | HelloApplication.class,args | ;
               }
       application.properties \Box \Box \Box \Box \Box \Box \Box
spring.application.name
eureka.client.serviceUrl.defaultZone
spring.application.name=hello-service
           eureka.client.serviceUrl.defaultZone=http://localhost:1
    111/eureka/
       service
                  П
                                                                    Tomcat
                         П
                                П
                                                                                             П
```

| $com.netflix.discovery.DiscoveryClient \verb $ |
|--|
| |
| s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat |
| started on port[]s[]: 8080[]http[] |
| c.n.e.EurekaDiscoveryClientConfiguration : Updating |
| port to 8080 |
| com.didispace.HelloApplication : Started |
| HelloApplication in 7.218 |
| seconds JVM running for 11.646 |
| com.netflix.discovery.DiscoveryClient : |
| DiscoveryClient_HELLO-SERVICE/PC- |
| 201602152056:hello-service-registration status: 204 |
| hello-service |
| |
| c.n.e.registry.AbstractInstanceRegistry : Registered |
| instance |
| HELLO-SERVICE/PC-201602152056:hello-service with |
| status UP□replication=true□ |
| Eureka Instances currently registered |
| with Eureka |
| Instances automatic registered with Funda |
| Instances currently registered with Eureka Application AMIs Availability Zones Status |
| HELLO-SERVICE n/a (1) (1) UP (1) - PC-201602152056:hello-service |
| http://localhost:8080/hello |
| |

```
com.didispace.web.HelloController
/hello,host:PC-201602152056,
  service id:hello-service
 Eureka Server
eureka.client.register-with-eureka=false
  eureka.client.fetch-registry=false
 oxed{1}
serviceUrl peer 2
  spring.application.name=eureka-server
  server.port=1111
  eureka.instance.hostname=peer1
  eureka.client.serviceUrl.defaultZone=http://peer2:111
2/eureka/
```

| ullet application-peer2.propertie | es peer2 |
|---|-----------------------------|
| serviceUrlpeer1_ | |
| spring.application.name=eure | ka-server |
| server.port=1112 | |
| eureka.instance.hostname=pe | er2 |
| eureka.client.serviceUrl.defaul | tZone=http://peer1:111 |
| 1/eureka/ | |
| ● /etc/hosts peer1 pe | er20000000 host |
| serviceUrl 🛛 🖺 🗎 🗎 🖺 🖺 | □Windows □ □ □ □ |
| C:\Windows\System32\drivers\etc\hc | osts[] |
| 127.0.0.1 peer1 | |
| 127.0.0.1 peer2 | |
| ●□□spring.profiles.active□□□□□□□ |]peer1[]peer2[] |
| java-jar | eureka-server-1.0.0.jar |
| spring.profiles.active=peer1 | |
| java-jar | eureka-server-1.0.0.jar |
| spring.profiles.active=peer2 | |
| <pre>Deer1DDDDhttp://localhost</pre> | ::1111/000000000000000 |
| registered-replicas [] [] peer 2 [] [] eu | reka-server[][][][][][][][] |
| peer2[[][http://localhost:1112/[[[|]registered-replicas |
| _peer1available- | replicase |
| peer1 http://localhost:1112/ | peer1 |
| □unavailable-replicas | |
| | |

| peer2 | | | |
|---|-----------|--------------------------|--|
| Instances curr | ently reg | gistered with E | ureka |
| Application | AMis | Availability Zones | Status |
| EUREKA-SERVER | n/a (2) | (2) | UP (2) - Lenovo-zhaiyc:eureka-server:1111 , Lenovo-zhaiyc:eureka-server:1112 |
| General Info | | | |
| Name | | | Value |
| total-avail-memory | | | 748mb |
| environment | | | test |
| num-of-cpus | | | 8 |
| current-memory-usage | | | 263mb (35%) |
| server-uptime | | | 00:00 |
| registered-replicas
unavailable-replicas | | | http://peer2:1112/eureka/ |
| available-replicas | | | http://peer2:1112/eureka/, |
| | - | erties∏∏∏
lication.na | ame=hello-service |
| • | • | | eUrl.defaultZone=http://peer1:111 |
| | | | 11 2/eureka/ |
| ПППППП |]∏∏eu | reka.clien | ıt.serviceUrl.defaultZone∏∏∏∏∏∏ |
| |
ПППП | eer1∏pee | |
| | | —. | er2∏ |
| | _ | пппп | |
| | | | 🛮 🔻 🖂 🗘 http://localhost:1111/ |
| ttp://local | □ □ □ | L112/000 | |
| ttp://local | □ □ □ | L112/000 | 🛮 🔻 🖂 🗘 http://localhost:1111/ |
| ttp://local
eer2[][[| | l112/0000
beer10000 | |
| ttp://local
eer2[][[[
eer2[][][] | | 112/ | |

DS Replicas

| | 100000000000000000000000000000000000000 |
|---------------------------|---|
| Spr | ing Boot |
| Eureka |]hello-service |
| | |
| Eureka |]RibbonRibbon |
| _HTTP_TCP |]ribbonServerList |
| | on 🛮 Eureka 🖺 🖺 Ribbon 🖺 🖺 🗎 |
| RibbonServerList [] Disco | veryEnabledNIWSServerList [][][][|
| Eureka | NIWSDiscoveryPing |
| IPing | Ribbon |
| Eureka | |
| | |
| | Eureka |
| ullet | |
| service | jar |
| hello-service | |
| java-jar | hello-service-0.0.1-SNAPSHOT.jar |
| server.port=8081 | |
| java-jar | hello-service-0.0.1-SNAPSHOT.jar |
| server.port=8082 | |
| ●□□□□□□ hello-service | e Eureka |
| HELLO-SERVICE | |
| 8082 | |
| | |

| Instances currently registered with Eureka | | | | | |
|--|---------|--------------------|--|--|--|
| Application | AMIs | Availability Zones | Status | | |
| HELLO-SERVICE | n/a (2) | (2) | UP (2) - PC-201602152056:hello-service:8081 , PC-201602152056:hello-service:8082 | | |

```
<parent>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-parent</artifactId>
 <version>1.3.7.RELEASE
 <relativePath/> <!-- lookup parent from repository -->
</parent>
<dependencies>
 <dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-web</artifactId>
 </dependency>
 <dependency>
   <groupId>org.springframework.cloud
   <artifactId>spring-cloud-starter-eureka</artifactId>
 </dependency>
 <dependency>
   <groupId>org.springframework.cloud</groupId>
   <artifactId>spring-cloud-starter-ribbon</artifactId>
 </dependency>
</dependencies>
<dependencyManagement>
 <dependencies>
   <dependency>
    <groupId>org.springframework.cloud
    <artifactId>spring-cloud-dependencies</artifactId>
    <version>Brixton.SR5</version>
    <type>pom</type>
    <scope>import</scope>
   </dependency>
 </dependencies>
</dependencyManagement>
```

| OnsumerApplication |
|--|
| @EnableDiscoveryClient |
| |
| @LoadBalanced |
| @EnableDiscoveryClient |
| @SpringBootApplication |
| public class ConsumerApplication { |
| @Bean |
| @LoadBalanced |
| RestTemplate restTemplate [[[] { |
| return new RestTemplate□□; |
| } |
| public static void main[String[]args[]{ |
| SpringApplication.run |
| □ConsumerApplication.class,args□; |
| } |
| } |
| ●□□ConsumerController□□□□/ribbon-consumer□□□□□□□□□ |
| □□□□□□□ RestTemplate □□□□ HELLO-SERVICE □□□□□/hello□□□ |
| |
| |
| @RestController |
| <pre>public class ConsumerController {</pre> |
| @Autowired |
| RestTemplate restTemplate; |
| <pre>@RequestMapping</pre> <pre></pre> |
| consumer",method=RequestMethod.GET[] |

```
public String helloConsumer[][{
    return restTemplate.getForEntity [] "http://HELLO-
SERVICE/hello",String.class[].getBody[][];
}
```

spring.application.name=ribbon-consumer
server.port=9000
eureka.client.serviceUrl.defaultZone=http://localhost:1
111/eureka/

| Instances currently r | egistered with | Eureka | |
|-----------------------|----------------|--------------------|--|
| Application | AMIs | Availability Zones | Status |
| HELLO-SERVICE | n/a (2) | (2) | UP (2) - PC-201602152056:hello-service/8081 , PC-201602152056:hello-service/8082 |
| RIBBON-CONSUMER | n/a (1) | (1) | UP (1) - PC-201602152056:ribbon-consumer:9000 |

c.n.l.DynamicServerListLoadBalancer
DynamicServerListLoadBalancer for client
HELLO-SERVICE initialized:

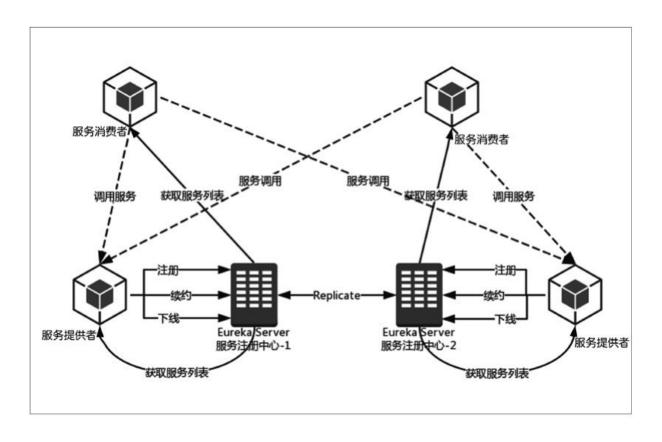
```
DynamicServerListLoadBalancer:
{NFLoadBalancer:name=HELLO-SERVICE,current list of
   Servers=[PC-201602152056:8082,PC-
201602152056:8081], Load balancer stats=Zone stats:
   {defaultzone=[Zone:defaultzone; Instance count:2;
Active connections count: 0:
   Circuit breaker tripped count: 0; Active connections per
server: 0.0;1
   },Server stats:[[Server:PC-201602152056:8082;
Zone:defaultZone; Total Requests:0;
   Successive connection failure:0; Total
                                               blackout
seconds:0;Last connection made:Thu Jan
   01 08:00:00 CST 1970; First connection made: Thu
Jan 01 08:00:00 CST 1970; Active
   Connections:0; total failure count in last ☐ 1000 ☐
msecs:0; average resp time:0.0; 90
   percentile resp time:0.0;
                                  95 percentile resp
time:0.0; min resp time:0.0; max resp
   time:0.0; stddev resp time:0.0]
   ,[Server:PC-201602152056:8081; Zone:defaultZone:
Total Requests:0; Successive
   connection failure:0; Total blackout seconds:0;Last
connection made:Thu Jan 01 08:00:00
   CST 1970; First connection made: Thu Jan 01
08:00:00 CST 1970; Active Connections:0;
   total failure count in last 1000 msecs:0; average resp
time:0.0; 90 percentile resp
```

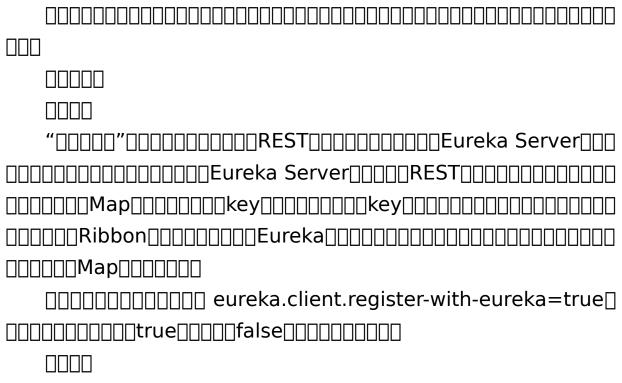
Eureka \square



| Eureka 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 💮 Eureka 🗆 🗆 💮 💮 Eureka 🗆 🗆 💮 Eureka 🗆 🗆 💮 Eureka 🗆 🖂 Eureka 🖂 🖂 Eureka |
|--|
| ●000000 Eureka000000000000000000000000000000000000 |
| eureka-server[] |
| ●□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| Eureka |
| |
| |
| RibbonFeign |
| |
| |
| Spring Cloud Eureka |
| EurekaEureka |
| |
| ●"□□□□□□-1"□"□□□□□□-2"□□□□□□□□□□□□□□□□ |
| ullet "00000"0000000000000000000000000000000 |
| □-2"□□ |

ullet

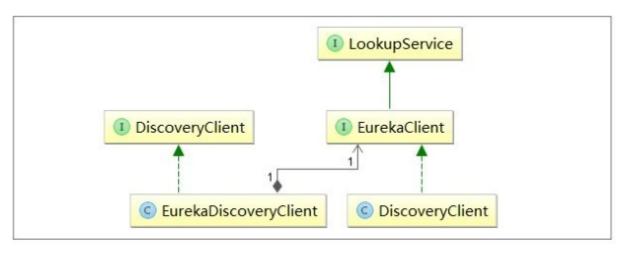


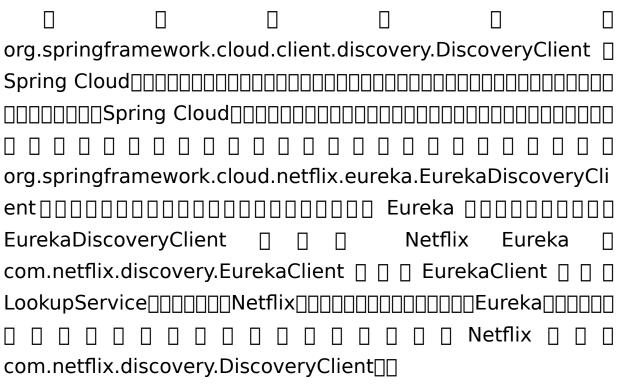


| Eureka Server:" |
|---|
| _ Eureka Server _" |
| □Renew□□ |
| |
| eureka.instance.lease-renewal-interval-in-seconds=30 |
| eureka.instance.lease-expiration-duration-in- |
| seconds=90 |
| eureka.instance.lease-renewal-interval-in-seconds [[][[] |
| $\ \ \square \square \square \square \square \square \square \square $ |
| expirationduration-in-seconds |
| |
| |
| |
| DDDD REST DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| |
| |
| $eureka.client.registry-fetch-interval-seconds=30 \verb $ |
| |
| |
| |
| |

| EurekaRegion_ZoneRegion |
|---|
| $Zone \verb $ |
| Zone |
| |
| |
| |
| |
| RESTEureka Server |
| |
| |
| |
| |
| Eureka |
| Server6060 |
| 90000000000 |
| |
| 00000000Eureka000000000000000000000000000000000000 |
| |
| EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING |
| INSTANCES ARE UP WHEN THEY'RE NOT.RENEWALS ARE |
| LESSER THAN THRESHOLD AND HENCE THE INSTANCES |
| ARE NOT BEING EXPIRED JUST TO BE SAFE. |
| 0000000000Eureka Server0000000000000000000000000000000000 |
| Eureka Server |
| Server85% |
| 00000000000000000000000000000000000000 |

| eureka.server.enableself-preservation=false |
|---|
| |
| |
| |
| 00000 Eureka 000000000000000000000000000000000000 |
| |
| |
| |
| 00000000000000000000000000000000000000 |
| Eureka |
| |
| |
| Server |
| ●□□□□□□□@EnableDiscoveryClient□□□ |
| ● □ application.properties □ □ |
| eureka.client.serviceUrl.defaultZone |
| |
| @EnableDiscoveryClient |
| @Target[]ElementType.TYPE[] |
| @Retention Retention Policy. RUNTIME |
| @Documented |
| |





```
Eureka Client
    -∏Eureka Server∏∏∏∏∏
    -∏Eureka Server∏∏∏
    -----Eureka Server
    -

-

Eureka Server
    Eureka Client□□□□□□□Eureka Server□URL□□□
  URL
                               П
                                  П
                                    П
eureka.client.serviceUrl.defaultZone
Dom.netflix.discovery.endpoint.EndpointUtils
public
              static
                       Map<String,List<String>>
  getServiceUrlsMapFromConfig[]
           EurekaClientConfig
                             clientConfig,String
        instanceZone,boolean
    preferSameZone∏{
      Map<String,List<String>>
                              orderedUrls=new
    LinkedHashMap<>□□;
      String region=getRegion

ClientConfig

;
      String[]availZones=clientConfig.getAvailabilityZon
    es∏clientConfig.getRegion∏∏;
      if \exists availZones == null \mid availZones.length == 0 \mid \{
        availZones=new String[1];
```

```
availZones[0]=DEFAULT ZONE;
       }
       int
                       myZoneOffset=getZoneOffset
    □instanceZone,preferSameZone,availZones□;
       String zone=availZones[myZoneOffset];
       List<String>
    serviceUrls=clientConfig.getEurekaServerServiceUrls
    []zone[];
       if∏serviceUrls !=null∏{
         orderedUrls.put□zone,serviceUrls□;
       }
       return orderedUrls:
   Region Zone
   ____Region___Zone___
_____Region______default______
eureka.client.region
    public static String getRegion [] EurekaClientConfig
 clientConfig∏{
      String region=clientConfig.getRegion□□;
     if∏region==null∏{
       region=DEFAULT_REGION;
      }
```

```
region=region.trim\square.toLowerCase\square;
     return region;
  ● 🛮 🗎 getAvailabilityZones 🖺 🖺 🖺 🖺 🖺 🖺 🖺 Region 🖺
eureka.client.availability-zones
public String[]getAvailabilityZones[String region[]{
     String value=this.availabilityZones.get region;
     if∏value==null∏{
      value=DEFAULT ZONE;
     return value.split∏","∏;
  serviceUrls
  int
                   myZoneOffset=getZoneOffset
 □instanceZone,preferSameZone,availZones□;
   String zone=availZones[myZoneOffset];
   List<String>
 serviceUrls=clientConfig.getEurekaServerServiceUrls
 ∏zone∏;
   □ □ □ serviceUrls
getEurekaServerServiceUrls
                   П
                     П
                        П
                           П
                             П
                                     П
```

| EurekaClientConfigBean | | | | Eure | kaCli | entC | onfig | |
|--|-------|-------|--------|-------|-------|------|-------|--|
| EurekaConstants [[] [[] [[] [[] [[] [[] [[] [[] [[] [[| | | | | | | | |
| | | | defa | ultZo | ne[[[| | | |
| 000000000000000 | | | | | | | | |
| eureka.client.serviceUrl.d | defau | ıltZo | ne 🛚 🗀 | | | | | |

```
public List<String> getEurekaServerServiceUrls(String myZone) {
    String serviceUrls = this.serviceUrl.get(myZone);
    if (serviceUrls == null || serviceUrls.isEmpty()) {
        serviceUrls = this.serviceUrl.get(DEFAULT_ZONE);
    }
    if (!StringUtils.isEmpty(serviceUrls)) {
        final String[] serviceUrlsSplit =
StringUtils.commaDelimitedListToStringArray(serviceUrls);
        List<String> eurekaServiceUrls = new ArrayList<> (serviceUrlsSplit.length);
        for (String eurekaServiceUrl : serviceUrlsSplit) {
            if (!endsWithSlash(eurekaServiceUrl)) {
                  eurekaServiceUrl += "/";
            }
                  eurekaServiceUrls.add(eurekaServiceUrl);
        }
        return eurekaServiceUrls;
    }
    return new ArrayList<>();
}
```

```
public void run() {
    try {
        discoveryClient.refreshInstanceInfo();
        Long dirtyTimestamp = instanceInfo.isDirtyWithTime();
        if (dirtyTimestamp != null) {
             discoveryClient.register();
             instanceInfo.unsetIsDirty(dirtyTimestamp);
        }
    } catch (Throwable t) {
        logger.warn("There was a problem with the instance info replicator", t);
    } finally {
        Future next = scheduler.schedule(this, replicationIntervalSeconds,
        TimeUnit.SECONDS);
        scheduledPeriodicRef.set(next);
    }
}
```

```
boolean register() throws Throwable {
     logger.info(PREFIX + appPathIdentifier + ": registering service...");
     EurekaHttpResponse<Void> httpResponse;
       httpResponse = eurekaTransport.registrationClient.register(instanceInfo);
     } catch (Exception e) {
       logger.warn("{} - registration failed {}", PREFIX + appPathIdentifier,
e.getMessage(), e);
      throw e;
     if (logger.isInfoEnabled()) {
       logger.info("{} - registration status: {}", PREFIX + appPathIdentifier,
httpResponse.getStatusCode());
     return httpResponse.getStatusCode() == 204;
   private void initScheduledTasks□□{
      if∏clientConfig.shouldFetchRegistry∏∏∏{
        //registry cache refresh timer
        int
     registryFetchIntervalSeconds=clientConfig.getRegistry
     FetchIntervalSeconds□□:
        int
     expBackOffBound=clientConfig.getCacheRefreshExecu
     torExponentialBackOffBound∏;
        scheduler.schedule∏
```

```
"cacheRefresh",
       scheduler,
       cacheRefreshExecutor,
       registryFetchIntervalSeconds,
       TimeUnit.SECONDS,
       expBackOffBound,
       \Box,
     registry Fetch Interval Seconds, Time Unit. SECONDS\\
 Π;
 }if∏clientConfig.shouldRegisterWithEureka∏∏{
   int
renewalIntervalInSecs=instanceInfo.getLeaseInfo
□□.getRenewalIntervalInSecs□□;
   int
expBackOffBound=clientConfig.getHeartbeatExecutorE
xponentialBackOffBound∏;
   logger.info

□"Starting heartbeat executor: "+"renew
interval is: "+renewalIntervalInSecs□;
   //Heartbeat timer
   scheduler.schedule□
     "heartbeat".
       scheduler,
       heartbeatExecutor,
       renewalIntervalInSecs,
       TimeUnit.SECONDS,
```

```
expBackOffBound,
       new HeartbeatThread□□
      Π,
      renewalIntervalInSecs,TimeUnit.SECONDS□;
     //InstanceInfo replicator
    }
  "_____if_____if______Eureka Server ______
eureka.instance.lease-renewal-interval-in-seconds=30
   eureka.instance.lease-expiration-duration-in-
 seconds=90
  n"nnnn"nnnnnnnn if nnnnnnnnnnnnnnnnn
eureka.client.fetch-registry=true [][][][][true[][][][][][][][][]
registryFetchIntervalSeconds | | | | | | | | | | | | | | | | |
30∏∏
  boolean renew∏∏{
    EurekaHttpResponse<InstanceInfo>httpResponse;
    try {
```

```
httpResponse=eurekaTransport.registrationClient.s
                            instanceInfo.getAppName
    endHeartBeat
    \square \square, instanceInfo.getId\square \square, instanceInfo, null\square;
       logger.debug
                      П
                           "{}-Heartbeat
                                           status:
     {}",PREFIX+appPathIdentifier,httpResponse.getStatus
    Code∏∏;
       if || httpResponse.getStatusCode || = = 404 || {
         REREGISTER COUNTER.increment□□;
         logger.info
                                  "{}-Re-registering
                         П
      apps/{}",PREFIX+appPathIdentifier,instanceInfo.getA
      ppName∏∏;
         return register□□;
        }
       returnhttpResponse.getStatusCode\square==200;
      } catch∏Throwable e∏{
       logger.error □ "{}-was unable to send heartbeat!
    ",PREFIX+appPathIdentifier,e∏;
        return false:
      }
   n"nnna"nnannannannannannan REST anaannannanna
com.netflix.eureka.resources
```

```
@POST
  @Consumes[]{"application/json","application/xml"}[]
  public Response addInstance Instance Info,
          @HeaderParam
      □PeerEurekaNode.HEADER REPLICATION□String
isReplication □ {
    logger.debug
                    П
                         "Registering
                                          instance
                                                      {}
\neg replication = \{\} \neg ", info.getId \neg \neg, \}
isReplication∏;
   //validate that the instanceinfo contains all the
necessary required fields
   //handle cases where clients may be registering
with bad DataCenterInfo with missing data
    DataCenterInfo
dataCenterInfo=info.getDataCenterInfo□□;
    if∏dataCenterInfo instanceof UniqueIdentifier∏{
      String dataCenterInfold= [] [] UniqueIdentifier []
  dataCenterInfo \square.getId \square \square;
      if∏isBlank∏dataCenterInfold∏∏{
        boolean
    experimental="true".equalsIgnoreCase
            serverConfig.getExperimental
        \square"registration.validation.
dataCenterInfold"□□:
        if∏experimental∏{
```

```
entity="DataCenterInfo of type
         String
     "+dataCenterInfo.getClass [] + " must contain a
     valid id";
                  Response.status ☐ 400 ☐ .entity
         return
     \sqcapentity\sqcap.build\sqcap\sqcap;
            else if □ dataCenterInfo
                                            instanceof
   AmazonInfo∏{
         AmazonInfo amazonInfo ☐ AmazonInfo ☐
     dataCenterInfo:
         String
                           effectiveId=amazonInfo.get

□AmazonInfo.MetaDataKey.instanceId□;

         if∏effectiveId==null∏{
           amazonInfo.getMetadata∏.put∏
               AmazonInfo.MetaDataKey.instanceId.ge
           tName \Pi, info.getId \Pi\Pi;
         }
        } else {
         logger.warn [] "Registering DataCenterInfo of
     type {} without an
appropriate id",
             dataCenterInfo.getClass□□□;
        }
      }
   registry.register∏info,"true".equals∏isReplication∏;
   return Response.status ☐ 204 ☐ .build ☐ ☐; //204 to be
backwards compatible
```

```
}
              П
                П
org.springframework.cloud.netflix.eureka.server.InstanceRegi
            П
                register
                          InstanceInfo
                                    info,int
stry
                       П
public
               register
                                    info,int
          void
                         InstanceInfo
 leaseDuration,boolean isReplication [{
      if□log.isDebugEnabled□□□{
        log.debug∏"register "+info.getAppName∏+",vip
     "+info.getVIPAddress□□
            +".leaseDuration
         "+leaseDuration+", isReplication "
            +isReplication□;
      }
      this.ctxt.publishEvent
                                      new
    EurekalnstanceRegisteredEvent this, info,
          leaseDuration,isReplication∏;
      super.register[info,leaseDuration,isReplication[];
    }
  _____ publishEvent __________
com.netflix.eureka.registry.AbstractInstanceRegistry [][][][]
_____ key _____InstanceInfo__
```



| 0000 Eureka 0000000000 Eureka 000000000000000000000000000000000000 | | | | | | |
|--|--|--|--|--|--|--|
| 00000000000000Eureka000000000000000000000000000000000000 | | | | | | |
| | | | | | | |
| _Eureka | | | | | | |
| | | | | | | |
| 0000000000000API 000000000000000Eureka | | | | | | |
| | | | | | | |
| EurekaEurekaEureka | | | | | | |
| DDDSpring Cloud Eureka | | | | | | |
| Eureka | | | | | | |
| Eureka | | | | | | |
| ullet | | | | | | |
| ●□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ | | | | | | |
| _Eureka | | | | | | |
| | | | | | | |
| org.springframework.cloud.netflix.eureka.server.EurekaServe | | | | | | |
| rConfigBean | | | | | | |
| | | | | | | |
| enableSelfPreservation | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| org.springframework.cloud.netflix.eureka.EurekaClientConfig | | | | | | |

```
eureka.client.serviceUrl
HashMap [][[][[][][][][][][][] key []defaultZone[]value []
http://localhost:8761/eureka/
          Map<String,String> serviceUrl=new
   private
 HashMap<>∏∏;
   {
     this.serviceUrl.put□DEFAULT ZONE,DEFAULT URL□;
   }
   public
                      final
                               String
             static
 DEFAULT URL="http://localhost:8761"+DEFAULT PREFIX+"/
   public
                      final
                               String
             static
 DEFAULT ZONE="defaultZone";
  Eureka∏∏∏∏
   eureka.client.serviceUrl.defaultZone=http://localhost:1
 111/eureka/
  eureka.client.serviceUrl.defaultZone=http://peer1:111
 1/eureka/,http://peer2:111 2/eureka/
```

| | | الملاكمة | | | | | |
|--|--|----------|---------------|--|------|--|--|
| S | serviceUrl 🛮 🗎 🗎 🗎 value 🖺 🗎 URL 🖺 🗎 🗎 🗎 🗎 🗎 🗎 🗎 🖺 🖺 🖺 🖺 | | | | | | |
| r | $http://{<}username>:{<}password>@localhost:1111/eureka \ [] \ []$ | | | | | | |
| <username><password></password></username> | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| org.spring framework.cloud.net flix.eureka. Eureka Client Config | | | | | | | |
| Bean [][[][[][][][][][][][][][][][][][][][] | | | | | | | |
| | | | | | | | |
| | 参数名 | | 说明 | | 默认值 | | |
| | enabled | | 启用 Eureka 客户端 | | true | | |

| 参数名 | 说明 | 默认值 |
|--|---|-------|
| registryFetchIntervalSeconds | 从 Eureka 服务端获取注册信息的间隔时间,
单位为秒 | 30 |
| instanceInfoReplicationIntervalSeconds | 更新实例信息的变化到 Eureka 服务端的间隔时间,单位为秒 | 30 |
| initial Instance Info Replication Interval Seconds | 初始化实例信息到 Eureka 服务端的间隔时间,单位为秒 | 40 |
| eurekaServiceUrlPollIntervalSeconds | 轮询 Eureka 服务端地址更改的间隔时间,单位为秒。当我们与 Spring Cloud Config 配合,动态刷新 Eureka 的 serviceURL 地址时需要关注该参数 | 300 |
| eurekaServerReadTimeoutSeconds | 读取 Eureka Server 信息的超时时间,单位为秒 | 8 |
| eurekaServerConnectTimeoutSeconds | 连接 Eureka Server 的超时时间,单位为秒 | 5 |
| eurekaServerTotalConnections | 从 Eureka 客户端到所有 Eureka 服务端的连接
总数 | 200 |
| eurekaServerTotalConnectionsPerHost | 从Eureka 客户端到每个Eureka 服务端主机的
连接总数 | 50 |
| eurekaConnectionIdleTimeoutSeconds | Eureka 服务端连接的空闲关闭时间,单位为秒 | 30 |
| heartbeatExecutorThreadPoolSize | 心跳连接池的初始化线程数 | 2 |
| heartbeatExecutorExponentialBackOffBound | 心跳超时重试延迟时间的最大乘数值 | 10 |
| cacheRefreshExecutorThreadPoolSize | 缓存刷新线程池的初始化线程数 | 2 |
| cacheRefreshExecutorExponentialBackOffBound | 缓存刷新重试延迟时间的最大乘数值 | 10 |
| useDnsForFetchingServiceUrls | 使用 DNS 来获取 Eureka 服务端的 serviceUrl | false |
| registerWithEureka | 是否要将自身的实例信息注册到 Eureka 服务端 | true |
| preferSameZoneEureka | 是否偏好使用处于相同 Zone 的 Eureka 服务端 | true |
| filterOnlyUpInstances | 获取实例时是否过滤,仅保留 UP 状态的实例 | true |
| fetchRegistry | 是否从 Eureka 服务端获取注册信息 | true |



| org. | sp | rin | gfr | am | ew | or | k.c | lou | d.n | etf | lix. | eu | rek | a.E | Eur | eka | alns | sta | nce | eCo | n |
|------|----|-----|-----|----|----|----|-----|-----|-----|-----|------|----|------|------|-----|------|------|-----|-----|-----|---|
| figB | ea | n [| | | | | | | | | eur | ek | a.ir | nsta | and | ce[] | | | | | |
| | | | | | | | | | | | | | | | | | | | | | |
| | ПП | П | | | | | | | | | | | | | | | | | | | |

| org.spring framework.cloud.net flix.eureka. Eureka Instance Connection (State 1998) and the connection of the connecti |
|--|
| figBean |
| Eureka |
| |
| |
| Spring Cloud Eureka |
| org.spring framework.cloud.net flix.eureka. Eureka Instance Connection (State 1988) and the connection of the connecti |
| figBean |
| $com.netflix.appinfo.InstanceInfo \verb $ |
| com.netflix.appinfo.InstanceInfo |
| ☐ Eureka ☐☐☐☐☐☐☐☐☐☐Map <string,string> metadata=new</string,string> |
| ConcurrentHashMap <string></string> |
| Spring Cloud _EurekalnstanceConfigBean |
| |
| <pre>Documentation ====================================</pre> |
| |
| |
| <key>=<value>[][][][][][][][][][][][][][][][][][][]</value></key> |
| eureka.instance.metadataMap.zone=shanghai |
| |
| |
| |
| Netflix Eureka |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |

| \${spring.cloud.client.hostname}:\${spring.application. |
|---|
| name}:\${spring.application.instance_id:\${server.port}} |
| |
| |
| server.port |
| |
| $server.port = \$\{random.int[10000,19999]\} \\ \verb $ |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| eureka.instance.instanceId=\${spring.application.name |
| <pre>}:\${random.int}}</pre> |
| |
| |
| |
| <pre>[]InstanceInfo[][][][][][][][]URL[][][][][]homePageUrl[]</pre> |
| $status Page Url \underline{\ \ } health Check Url \underline{\ \ \ } \underline{\ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ } \underline{\ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ } \underline{\ \ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ \ } \underline{\ \ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ \ } \underline{\ \ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ } \underline{\ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ } \underline{\ \ \ \ \ } \underline{\ \ \ \ \ } \ \ \ \$ |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| $spring-boot-actuator \verb $ |
| |
| Eureka DDDD/health DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| $\verb $ |
| /infoEureka |
| |
| 0000000000000000URL00000000000000000000 |
| $context-path \verb $ |
| /info |
| management context-nath=/hello |

```
eureka.instance.statusPageUrlPath=${management.co
 ntext-path}/info
   eureka.instance.healthCheckUrlPath=${management.c
 ontext-path}/health
  endpoints.info.path=/appInfo
   endpoints.health.path=/checkHealth
   eureka.instance.statusPageUrlPath=/${endpoints.info.
 path}
   eureka.instance.healthCheckUrlPath=/${endpoints.hea
 lth.path}
  eureka.instance.healthCheckUrlPath
<u>________Eureka______HTTP________</u>
Spring Cloud Eureka
   eureka.instance.statusPageUrl=https://${eureka.instan
 ce.hostname}/info
   eureka.instance.healthCheckUrl=https://${eureka.insta
 nce.hostname}/health
   eureka.instance.homePageUrl=https://${eureka.instan
 ce.hostname}/
  DDDDDEureka
```

| Spring Cloud Eureka |
|--|
| $\verb spring-boot-actuator \verb $ |
| |
| ●□pom.xml□□□spring-boot-starter-actuator□□□□□□□ |
| |
| ● □ application.properties □ □ □ □ □ □ |
| <pre></pre> |
| |
| eureka.client.healthcheck.enabled=true[] |
| eureka.client.healthcheck.enabled=true[] •[][][][][health[][][][][][][][][][][][][][][][][][][] |
| eureka.client.healthcheck.enabled=true[] •[][][][][][health[][][][][][][][][][][][][][][][][][][] |
| eureka.client.healthcheck.enabled=true[] •[][][][][][health[][][][][][][][][][][][][][][][][][][] |

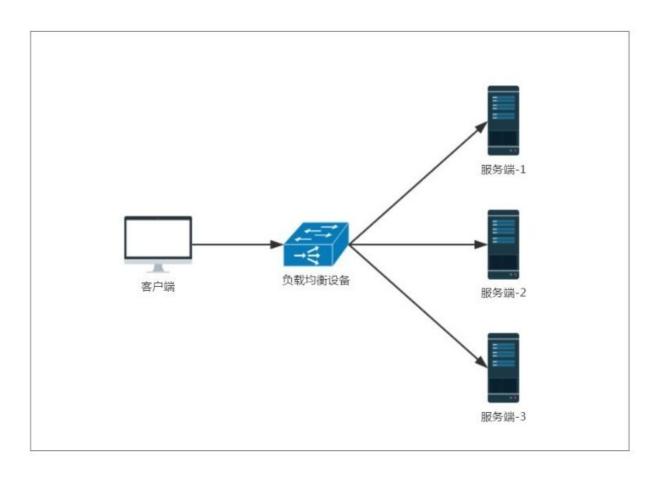
| 参数名 | 说明 | 默认值 |
|----------------------------------|---------------------------------------|-------|
| preferIpAddress | 是否优先使用 IP 地址作为主机名的标识 | false |
| leaseRenewalIntervalInSeconds | Eureka 客户端向服务端发送心跳的时间间隔,单位为秒 | 30 |
| leaseExpirationDurationInSeconds | Eureka 服务端在收到最后一次心跳之后等待的时间上限, | 90 |
| | 单位为秒。超过该时间之后服务端会将该服务实例从服务 | |
| | 清单中剔除,从而禁止服务调用请求被发送到该实例上 | |
| nonSecurePort | 非安全的通信端口号 | 80 |
| securePort | 安全的通信端口号 | 443 |
| nonSecurePortEnabled | 是否启用非安全的通信端口号 | true |
| securePortEnabled | 是否启用安全的通信端口号 | |
| appname | 服务名,默认取 spring.application.name 的配置值, | |
| | 如果没有则为 unknown | |
| hostname | 主机名,不配置的时候将根据操作系统的主机名来获取 | |

| "Eureka " |
|--|
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Eureka |
| Java |
| |
| |

| Eureka |
|--|
| Java |
| |
| |
| |
| js-client python-eureka nondered python-eurek |
| |
| Eureka 🛮 REST API 🖺 🖺 🖺 Eureka 🖺 WiKi 🗎 🖺 Eureka-REST- |
| operations $\ \square\ \square$ https://github.com/Netflix/eureka/wiki/Eureka- |
| REST-operations [|
| |
| Eureka Jersey XStream JSON Server Client |
| |
| ●Jersey[]JAX-RS[][][][][][][][][][][][][][][][][][][] |
| ■□□□□□ □Core Server□□□□□□ JSR 311 □□□□□□□ API □□□□□ |
| OOOOOOO RESTful Web |
| ■□□□□□ □Core Client□:Jersey □□□ API □□□□ REST □□□□□□□ |
| ■□□ □Integration□:Jersey □□□□□□□□□□□ Spring□Guice□ |
| Apache Abdera □□□ |

| ●XStream[][][][][][]XML[]JSON[][][][][][][][][]Java[][] |
|--|
| XStream |
| XStreamprivate_final |
| XStreamX |
| XMLXStreamXStreamXStream |
| XStream |
| |
| JAX-RS [][] Java EE 6[][][][][][][]JAX-RS [] Java API for |
| RESTful Web Services |
| _RESTWebJAX-RSJava SE 5JavaWeb_ |
| |
| ●@Path□□□□□□□□□□□□□ |
| ●@GET[]@PUT[]@POST[]@DELETE[][][][][]HTTP[][][][] |
| ●@Produces□□□□□MIME□□□□□ |
| ●@Consumes[[][[][][]MIME[][][] |
| @PathParam [] @QueryParam [] @HeaderParam [] |
| @CookieParam[]@MatrixParam[] |
| @FormParam[][][][][][]HTTP[][][][][][@PathParam |
| URL @QueryParam URL @HeaderParam |
| HTTP[][][][@CookieParam[]HTTP[][Cookie[] |
| 0000Eureka Server |
| |





CONTINUENTE DE LA CONSULA DE CONSULAU CONSULAUS CONSULAU

 CONTINUENTE DE CONTINUENTE DE CONSULAU CONSULAU

 CONTINUENTE DE CONTIN

| toConfiguration |
|--|
| |
| |
| ●□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| |
| |
| RestTemplate |
| |
| 3 🛮 🗎 " 🔻 🔻 🔻 🔻 🔻 🖺 🖺 O O O O O O O O O O O O O O O O O |
| DDDDDDDDDDDDDDDDD RestTemplate |
| RestTemplate |
| <u>GET∏</u> |
| RestTemplate GET |
| |
| |
| HTTP |
| USER-SERVER |
| ResponseEntity[][][]body[][][][][][][][][][][][][][][][][][][] |
| |

```
RestTemplate restTemplate=new RestTemplate□□;
                        ResponseEntity<String>
        responseEntity=restTemplate.getForEntity \( \precedit \) "http://USER-
        SERVICE/user? name={1}",String.class,"didi"[];
                        String body=responseEntity.getBody□□;

  \prod_{n=1}^{n} body = 0

                         RestTemplate restTemplate=new RestTemplate□□;
                        ResponseEntity<User>
        responseEntity=restTemplate.getForEntity ☐ "http://USER-
        SERVICE/user? name={1}",User.class,"didi"∏;
                        User body=responseEntity.getBody∏;
                nnnnnnnnnnngetForEntitynnnnnnnnnnnnnnnnnn
                                                   getForEntity
                                                                                                                               П
                                                                                                                                                                String
                                                                                                                                                                                                                      url.Class
responseType,Object...urlVariables
name=didi______u___url____url____url____url_____
urlVariables | | | | GET | | | | | | | | url | | | | http://USER-
SERVICE/user? name=\{1\} \square \square \square \square \square \square \square \square getForEntity
                                                       "http://USER-SERVICE/user?
П
                                                                                                                                                                                                                           name=
 url,Class responseType,Map
                 ■ getForEntity 

String
urlVariables
\sqcap \sqcap \mathsf{http://USER\text{-}SERVICE/user?} \quad \mathsf{name} = \{\mathsf{name}\} \sqcap \sqcap \mathsf{Map} \sqcap \sqcap \sqcap \sqcap \mathsf{Map} \sqcap \mathsf{map
```

```
RestTemplate restTemplate=new RestTemplate□□;
   Map<String> params=new HashMap<>□□;
   params.put[]"name","dada"[];
   ResponseEntity<String>
 responseEntity=restTemplate.getForEntity [] "http://USER-
 SERVICE/user? name={name}",String.class,params[];
  RestTemplate restTemplate=new RestTemplate□□;
   UriComponents
 "http://USER-SERVICE/user? name={name}"□
     .build∏∏
     .expand∏"dodo"∏
     .encode∏∏;
   URI uri=uriComponents.toUri∏;
   ResponseEntity<String>
 responseEntity=restTemplate.getForEntity
 □uri,String.class□.getBody□□;
  Description:
HttpMessageConverterExtractor HTTP HTTP HTTP Body HTTP HTTP
RestTemplate restTemplate=new RestTemplate□□;
```

| String | result=restTemplate.getForObject |
|-----------------------------------|---|
| <pre>[]uri,String.class[];</pre> | |
| body User | |
| RestTemplate rest | Template=new RestTemplate□□; |
| User result=restTe | emplate.getForObject[]uri,User.class[]; |
| body | Response |
| body getForEr | ntity |
| getForObject | ct String url,Class |
| responseType,Objectu | rlVariables |
| responseType | eurlVariablesurl |
| | |
| getForObject [] | String url,Class responseType,Map |
| urlVariables 🛮 🗎 🗎 🗎 🖺 🖺 | Map |
| urlVariables | Mapkey |
| ●getForObject URI | url,Class responseType[][][][][] URI [] |
| urlVariables | |
| | DOCTOR! |
| | <u>POST∏</u> |
| ∏RestTemplate∏∏P(| OSTODODODODODODO |
| |]
]GETgetForEntity |
| | |
| postForEntity[][POST[][[|]USER-SERVICE[[[]]/user[[[]][[]]body[] |
| user body[| □□□String□ |
| RestTemplate re | stTemplate=new RestTemplate[]; |
| User user=new | User□"didi",30□; |
| ResponseEntity< | <string> responseEntity=</string> |

| restTemplate.po | stForEntity | | "http://USER- |
|---|--------------------|----------------|----------------------|
| SERVICE/user",user, | ,String.class[|]; | |
| String body=r | responseEnti | ty.getBody∏[|]; |
| postForEntity | | | |
| postForEntity | String | url,Object | request,Class |
| response Type, Object. | uriVariables | 5 🗆 | |
| postForEntity | String | url,Object | request,Class |
| responseType,Map uri | Variables <u>□</u> | | |
| postForEntity | □ URI | url,Object | request,Class |
| responseType[] | | | |
| | getForEntity | | |
| uriVariables | url |][[[[][respons | seType[][][][][] |
| body | red | quest[][][][[| |
| HttpEntity | 0000000 | HttpE | ntity [] [] [] [] [] |
| RestTemplate [] [] [] [|]Http | Entity 🛮 🗎 🗎 🖺 |][[]Object[][|
| request [] [] [] request | | body | request |
| HttpEntity[][][][][][[][[|]http [|)00000000 r | equest 🛮 🖺 🖺 🖺 🗎 |
| body [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[| r | | |
| <pre>postForObject</pre> | :t | getForObject | |
| postForEntity | | body | |
| | | | |
| RestTemplate re | estTemplate= | new RestTer | nplate <u>□</u> []; |
| User user=new | User∐"didi", | 20□; | |
| String | postResult= | restTemplate | e.postForObject |
| □"http://USER-SERV | ICE/user",use | er,String.clas | s <u></u> ; |
| postForObject□□□ | | | |

| postForObject String url,Object request,Cla | SS | | | | | | | |
|--|-----|--|--|--|--|--|--|--|
| responseType,ObjecturiVariables[] | | | | | | | | |
| postForObject | SS | | | | | | | |
| responseType,Map uriVariables[| | | | | | | | |
| postForObject | SS | | | | | | | |
| responseType[] | | | | | | | | |
| postForEntity | | | | | | | | |
| postForEntity[][][| | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| User user=new User[]"didi",40[]; | | | | | | | | |
| URI responseURI=restTemplate.postForLocation | วท | | | | | | | |
| □"http://USER-SERVICE/user",user□; | | | | | | | | |
| postForLocation | o-t | | | | | | | |
| postForLocation [String url,ObjecturlVariables] | CL | | | | | | | |
| postForLocation [] String url,Object request,Ma | an | | | | | | | |
| urlVariables∏ | дρ | | | | | | | |
| ●postForLocation URI url, Object request □ | | | | | | | | |
| postForLocation | | | | | | | | |
| Description of the postfor of the po | | | | | | | | |
| responseType DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD | | | | | | | | |
| | | | | | | | | |
| <u>PUT</u> ∏∏ | | | | | | | | |
| RestTemplate PUT put | | | | | | | | |
| RestTemplate restTemplate=new RestTemplate[]; | | | | | | | | |
| | | | | | | | | |

| Long id=10001L; | | | | | | | | |
|---|------------------|-------------------------|--|--|--|--|--|--|
| User user=new User∏"didi | i",40 <u>□</u> ; | | | | | | | |
| restTemplate.put | | "http://USER- | | | | | | |
| SERVICE/user/{1}",user,id□; | | | | | | | | |
| put[][][][][][][][][][][][][][][][][][][] | | | | | | | | |
| ●put String url, Object reque | est,Obje | cturlVariables[] | | | | | | |
| ●put String url, Object reque | est,Map | urlVariables[] | | | | | | |
| ●put URI url,Object request | : | | | | | | | |
| put[][][void[][][][][][][][][] | |][[]responseType[][][] | | | | | | |
| postForObjec | :t | | | | | | | |
| | | | | | | | | |
| <u>DELE1</u> | | | | | | | | |
| <pre>□RestTemplate□□□DELETE□□</pre> |][[[]]del | ete | | | | | | |
| RestTemplate restTemplate | e=new I | RestTemplate <u>∏</u> ; | | | | | | |
| Long id=10001L; | | | | | | | | |
| restTemplate.delete | | "http://USER- | | | | | | |
| SERVICE/user/{1}",id□; | | | | | | | | |
| delete | | | | | | | | |
| ●delete String url, Objecturl Variables | | | | | | | | |
| ●delete String url, Map urlVariables | | | | | | | | |
| ●delete[]URI url[] | | | | | | | | |
| | ELETE | url | | | | | | |
| DELETE[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[| |)url | | | | | | |
| DELETEurlVariablesurl[| | | | | | | | |



| SpringRibbon |
|---|
| |
| |
| DDDDDDDDDDD@LoadBalancedDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| <pre>[@LoadBalanced [][][][][][][][][][][][][] RestTemplate []</pre> |
| LoadBalancerClient |
| LoadBalancerClient Spring Cloud |
| <pre>public interface LoadBalancerClient {</pre> |
| ServiceInstance choose[String serviceId[]; |
| <t> T execute String</t> |
| serviceId,LoadBalancerRequest <t> request [] throws</t> |
| IOException; |
| URI reconstructURI [] ServiceInstance instance,URI |
| original□; |
| } |
| |
| ◆ServiceInstance choose String serviceId □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| serviceId[][][][][][][][][][][][][] |
| ■ T execute [] String serviceId,LoadBalancerRequest |
| request[]throws IOException[][][][][][][][][][][][][][][][][][][] |
| URI reconstructURI |
| |

| host URI host:port | | | | | | | | |
|---|--|--|--|--|--|--|--|--|
| http://myservice/path/to/service | | | | | | | | |
| ServiceInstance | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| □ □ LoadBalancerClient □ □ □ □ | | | | | | | | |
| org.springframework.cloud.client.loadbalancer | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| ClientHttpRequestInterceptor LoadBalancerClient | | | | | | | | |
| 1 | | | | | | | | |
| © ConditionalOnBean © ConditionalOnClass © Configuration © LoadBalancerInterceptor | | | | | | | | |
| | | | | | | | | |
| © LoadBalancerAutoConfiguration | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD | | | | | | | | |
| | | | | | | | | |
| @Configuration | | | | | | | | |
| <pre>@ConditionalOnClass[]RestTemplate.class[]</pre> | | | | | | | | |
| <pre>@ConditionalOnBean[LoadBalancerClient.class[]</pre> | | | | | | | | |
| <pre>public class LoadBalancerAutoConfiguration {</pre> | | | | | | | | |
| @LoadBalanced | | | | | | | | |
| @Autowired required = false | | | | | | | | |

private

@Bean

 $restTemplates = Collections.emptyList \verb|||||;$

List<RestTemplate>

```
public
                             SmartInitializingSingleton
loadBalancedRestTemplateInitializer□
                       List<RestTemplateCustomizer>
       final
   customizers∏{
     return new SmartInitializingSingleton□□{
       @Override
       public void afterSingletonsInstantiated□□{
                    RestTemplate
                                    restTemplate
     LoadBalancerAutoConfiguration.
this.restTemplates[]{
           for 

☐ RestTemplateCustomizer customizer :
       customizers∏{
             customizer.customize restTemplate;
         }
       }
     };
   @Bean
   @ConditionalOnMissingBean
   public
                              RestTemplateCustomizer
restTemplateCustomizer[]
       final
                              LoadBalancerInterceptor
   loadBalancerInterceptor [ {
     return new RestTemplateCustomizer□□{
       @Override
```

```
RestTemplate
      restTemplate[]{
          List<ClientHttpRequestInterceptor> list=new
       ArrayList<>□
            restTemplate.getInterceptors[[]];
          list.add∏loadBalancerInterceptor∏;
          restTemplate.setInterceptors[list];
         }
       };
      }
      @Bean
      public LoadBalancerInterceptor ribbonInterceptor

☐
         LoadBalancerClient loadBalancerClient | {
                       LoadBalancerInterceptor
       return
                new
     □loadBalancerClient□;
      }
     }
  @ConditionalOnClass
● @ConditionalOnBean □ LoadBalancerClient.class □ □ □
Spring Bean CoadBalancer Client Bean
```

customize □

public

void

| ●□□□□□RestTemplateCustomizer□Bean□□□□RestTemplate |
|--|
| <pre>□□LoadBalancerInterceptor□□□□</pre> |
| ●□□□□□@LoadBalanced□□□□□RestTemplate□□□□□□□□□□ |
| |
| RestTemplate LoadBalancerInterceptor Description Des |
| |
| RestTemplate DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |

```
public class LoadBalancerInterceptor implements ClientHttpRequestInterceptor {
   private LoadBalancerClient loadBalancer;
   public LoadBalancerInterceptor(LoadBalancerClient loadBalancer) {
      this.loadBalancer = loadBalancer;
   @Override
   public ClientHttpResponse intercept(final HttpRequest request, final byte[] body,
         final ClientHttpRequestExecution execution) throws IOException {
      final URI originalUri = request.getURI();
      String serviceName = originalUri.getHost();
      return this.loadBalancer.execute(serviceName,
            new LoadBalancerRequest<ClientHttpResponse>() {
                public ClientHttpResponse apply(final ServiceInstance instance)
                      throws Exception {
                   HttpRequest serviceRequest = new ServiceRequestWrapper(request,
                         instance);
                   return execution.execute(serviceRequest, body);
                }
             });
   private class ServiceRequestWrapper extends HttpRequestWrapper {
      private final ServiceInstance instance;
      public ServiceRequestWrapper(HttpRequest request, ServiceInstance instance) {
         super (request);
         this.instance = instance;
      @Override
      public URI getURI() {
         URI uri = LoadBalancerInterceptor.this.loadBalancer.reconstructURI(
                this.instance, getRequest().getURI());
         return uri;
```



| RestTemplate |
|--|
| getHost |
| LoadBalancerClient |
| Ribbon |
| org.springframework.cloud.netflix.ribbon 🛮 🗎 🗎 🗎 🗎 🗎 🗎 |
| RibbonLoadBalancerClient[] |

```
public <T> T execute(String serviceId, LoadBalancerRequest<T> request) throws
IOException {
       ILoadBalancer loadBalancer = getLoadBalancer(serviceId);
       Server server = getServer(loadBalancer);
       if (server == null) {
          throw new IllegalStateException("No instances available for " + serviceId);
       RibbonServer ribbonServer = new RibbonServer(serviceId, server,
isSecure(server,
             serviceId), serverIntrospector(serviceId).getMetadata(server));
       RibbonLoadBalancerContext context = this.clientFactory
             .getLoadBalancerContext(serviceId);
       RibbonStatsRecorder statsRecorder = new RibbonStatsRecorder(context, server);
       try {
          T returnVal = request.apply(ribbonServer);
          statsRecorder.recordStats(returnVal);
          return returnVal;
       catch (IOException ex) {
          statsRecorder.recordStats(ex);
          throw ex;
       catch (Exception ex) {
          statsRecorder.recordStats(ex);
          ReflectionUtils.rethrowRuntimeException(ex);
       return null;
```

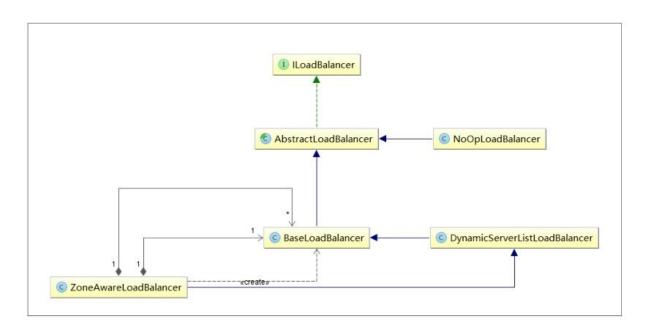
```
getServer ☐ ILoadBalancer
               protected
                                                Server
     loadBalancer∏{
                    if∏loadBalancer==null∏{
                         return null:
                    }
                    return loadBalancer.chooseServer□"default"□;
          LoadBalancerClient \( \propto 
ILoadBalancer□□□□□□chooseServer□□□
          public interface ILoadBalancer {
                    public void addServers□List<Server> newServers□;
                    public Server chooseServer

☐ Object key
☐;
                    public void markServerDown

∏Server server

∏;
                    public List<Server> getReachableServers□□;
                    public List<Server> getAllServers□□;
               }
          •chooseServer
          •getAllServers
          ____host_port_____
```

| | | | | | | | | | | Dynamic Server List Load Balancer | |
|---|--|--|--|--|--|--|--|--|--|-----------------------------------|--|
| ZoneAwareLoadBalancer 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 | | | | | | | | | | | |



@Bean

@ConditionalOnMissingBean

public ILoadBalancer ribbonLoadBalancer [IClientConfig config,

ServerList<Server>

serverList,ServerListFilter<Server> serverListFilter,

IRule rule,IPing ping[[{

ZoneAwareLoadBalancer<Server>

balancer=LoadBalancerBuilder.newBuilder

| | . with Client Config | | config | | .withRule |
|-------|---------------------------------------|--------|-------------------|--------|-------------|
| | <pre>□rule□.withPing□ping□</pre> | | | | |
| | .withServerListFilter | | | | |
| | ☐serverListFilter ☐.withDy | /nam | icServer | List∏s | erverList[] |
| | .buildDynamicServe | rListl | _oadBala | ncer | l□; |
| | return balancer; | | | | |
| | } | | | | |
| | □□□□□□□ RibbonLoadBaland | cerCl | ient 🛮 ex | ecute | : 0000000 |
| Zon | neAwareLoadBalancer 🛮 choo | seSe | rver 🛮 🖺 🖸 | | |
| | ServerRibbons | Serve | er | | |
| |]serviceIdF | HTTP: | S00000 | | 0000000 |
| Load | dBalancerInterceptor[][][][|]Load | dBalance | erRequ | ıest∏apply |
| ∏fina | al ServiceInstance instance[| | | | |
| | host URI host:post | | | | |
| | ☐ apply ☐ final ServiceInsta | ance | instanc | e 🛮 🖺 | |
| Serv | viceInstance[][][][][][][][][][[][| | 1000000 | | |
| | serviceId_host_p | ort[[| | | |
| | public interface ServiceIns | stanc | e { | | |
| | String getServiceId□□; | | | | |
| | String getHost□□; | | | | |
| | int getPort <u>□</u> ; | | | | |
| | boolean isSecure□□; | | | | |
| | URI getUri□□; | | | | |
| | Map <string,string> g</string,string> | etMe | tadata <u> </u> [|]; | |
| | } | | | | |
| | Server _ | | ☐ Ribbo | onSer | ver 🛮 🖺 🖺 |
| Serv | viceInstance∏∏∏∏∏∏∏∏∏∏∏ | ⊓Se | rver∏∏∏ | וחחחחו | |

```
static class RibbonServer implements
    protected
 ServiceInstance {
       private final String serviceld;
       private final Server server;
       private final boolean secure;
       private Map<String,String> metadata;
       protected RibbonServer ☐ String serviceId, Server
    server∏{
         this
                     serviceld, server, false, Collections.
      <String, String> emptyMap∏∏;
       }
       protected RibbonServer ☐ String serviceId, Server
    server, boolean secure,
            Map<String,String> metadata∏{
         this.serviceId=serviceId;
         this.server=server:
         this.secure=secure:
         this.metadata=metadata;
       }
       //DDDServiceInstanceDDDDServerDDgetDD
   reconstructURI
    @Override
```

```
public ClientHttpResponse apply ☐final ServiceInstance
 instance□
           throws Exception {
       HttpRequest
                             serviceRequest=new
    ServiceRequestWrapper[]request,instance[];
       return execution.execute □ service Request, body □;
    }
     getURI 🛮 🗎 🗎 🗎 🗎 getURI 🖺 🖺 🗎 LoadBalancerClient 🖺 🗎 🗎
reconstructURI
    private
                           ServiceRequestWrapper
                class
 extendshttpRequestWrapper {
       private final ServiceInstance instance;
       @Override
       public URI getURI∏∏{
        URI
     uri=LoadBalancerInterceptor.this.loadBalancer.recon
     structURI<sub>□</sub>
             this.instance,getRequest□□.getURI□□□;
        return uri;
       }
    }
      LoadBalancerInterceptor
  П
```

```
□serviceRequest,body□□□□□□ InterceptingClientHttpRequest□
public
            ClientHttpResponse execute ☐ HttpRequest
 request,byte[]body[]throws IOException {
       if this.iterator.hasNext [ ] {
         ClientHttpRequestInterceptor
     nextInterceptor=this.iterator.next□□;
         return
                                 nextInterceptor.intercept
     ∏request,body,this∏;
       }
       else {
         ClientHttpRequest
     delegate=requestFactory.createRequest
     □request.getURI□□,request.getMethod□□□;
         delegate.getHeaders \( \property \).putAll \( \property \) request.getHeaders
     \Pi\Pi\Pi;
         if\sqcapbody.length > 0\sqcap{
          StreamUtils.copy\u00edbody,delegate.getBody\u00edl\u00dd\u00edl;
         }
         return delegate.execute∏;
       }
   □ □ □ □ □ □ □ □ □ □ □ requestFactory.createRequest
☐ request.getURI ☐ ☐ ,request.getMethod ☐ ☐ ☐ ; ☐ ☐ ☐
request.getURI [ ] [ ] [ ] [ ] ServiceRequestWrapper [ ] [ ] [ ] [
reconstructURI
```

```
instance,URI original□{
       Assert.notNull [] instance, "instance can not be
    null"∏;
       String serviceId=instance.getServiceId□□;
       RibbonLoadBalancerContext
    context=this.clientFactory
           .getLoadBalancerContext[]serviceId[];
                          Server ☐ instance.getHost
        Server
              server=new
    \sqcap \sqcap, instance.getPort\sqcap \sqcap \sqcap;
       boolean secure=isSecure∏server,serviceId∏;
       URI uri=original;
       if∏secure∏{
         uri=UriComponentsBuilder.fromUri ☐ uri ☐ .scheme
      \sqcap"https"\sqcap.build\sqcap\sqcap.toUri\sqcap\sqcap;
        }
                     context.reconstructURIWithServer
       return
    □server,uri□;
    }
   preconstructURI [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] ServiceInstance [ ] [ ] [ ] [ ]
RibbonLoadBalancerContext[][]reconstructURIWithServer[][]
RibbonLoadBalancerContext□
```

public

URI

reconstructURI □ ServiceInstance

```
■ RibbonLoadBalancerContext □ □ LoadBalancerContext □ □
API \sqcap \sqcap
⊓reconstructURIWithServer⊓⊓⊓⊓⊓⊓
  \squareServiceInstance\squarereconstructURIWithServer\square
□□Server□□□□RibbonLoadBalancerClient□□reconstructURI □□
reconstructURIWithServer
public
                            implements
              LoadBalancerContext
         class
 IClientConfigAware {
    public
             reconstructURIWithServer
         URI
                               Server
  server,URI original<sub>□</sub>{
     String host=server.getHost□□;
     int port=server.getPort□□;
     if | host.equals | original.getHost | | | |
        && port==original.getPort\sqcap \sqcap \sqcap \uparrow
      return original;
     String scheme=original.getScheme∏;
     if∏scheme==null∏{
```

```
scheme=deriveSchemeAndPortFromPartialUri
[]original[].first[][];
 }
 try {
   StringBuilder sb=new StringBuilder□□;
   sb.append\\\ scheme\\\\ .append\\\"://\\\\;
                             Strings.isNullOrEmpty
   if
□original.getRawUserInfo□□□□{
     sb.append original.getRawUserInfo \\n\\.append
 □"@"□;
    }
   if |port>=0|
     sb.append[]":"[].append[]port[];
   sb.append□original.getRawPath□□□;
   if∏! Strings.isNullOrEmpty∏original.getRawQuery
sb.append□"? "□.append□original.getRawQuery
 }
   if
                             Strings.isNullOrEmpty
□original.getRawFragment□□□□{
     sb.append
                                          .append
                                    □original.getRawFragment□□□;
    }
   URI newURI=new URI□sb.toString□□□;
```

```
return newURI;
                                              } catch□URISyntaxException e□{
                                                      throw new RuntimeException∏e∏;
                                              }
                                     }
                  On the state of th
host | URI |
Ribbon \square Ribbon Load Balancer Client \square \square \square \square \square \square Ribbon \square \square \square \square \square \square
ZoneAwareLoadBalancer
                                                                                                                             חחחחחחחחחחחחח Spring Cloud חחחח Ribbon חחחחחחחחחח
Ribbon DD RibbonLoadBalancerClient
```

AbstractLoadBalancer

```
•ALL
  ●STATUS UP□□□□□□□□□
  •STATUS_NOT_UP_____
  _____ chooseServer ___ ___ ___ ______
key[][][]
  ●getLoadBalancerStats□□□□□□□LoadBalancerStats□□□□
UnloadBalancerStats
public abstract class AbstractLoadBalancer implements
 ILoadBalancer {
    public enum ServerGroup{
     ALL,
     STATUS UP,
     STATUS_NOT_UP
    public Server chooseServer∏{
     return chooseServer □ null □;
    }
    public
                       getServerList
         abstract
               List<Server>
  GrverGroup serverGroup
```

```
public
                       LoadBalancerStats
             abstract
   getLoadBalancerStats□□;
  BaseLoadBalancer
  BaseLoadBalancer Ribbon Ribbon Ribbon
@Monitor
 □ name=PREFIX+"AllServerList",type=DataSourceType.INF
 ORMATIONAL
   protected
               volatile
                         List<Server>
 allServerList=Collections
     .synchronizedList new ArrayList < Server > □□□;
   @Monitor
 □ name=PREFIX+"UpServerList",type=DataSourceType.INF
 ORMATIONAL
               volatile
   protected
                         List<Server>
 upServerList=Collections
     .synchronizedList new ArrayList < Server > □□□;
  ullet
LoadBalancerStats□□□
  null_____
```

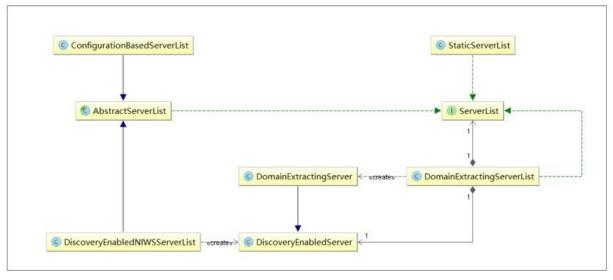
```
private static class SerialPingStrategy implements IPingStrategy {
  @Override
   public boolean[] pingServers(IPing ping, Server[] servers) {
      int numCandidates = servers.length;
      boolean[] results = new boolean[numCandidates];
      if (logger.isDebugEnabled()) {
         logger.debug("LoadBalancer: PingTask executing ["
                   + numCandidates + "] servers configured");
      for (int i = 0; i < numCandidates; i++) {
         results[i] = false;
         try {
            if (ping != null) {
               results[i] = ping.isAlive(servers[i]);
         } catch (Throwable t) {
           logger.error("Exception while pinging Server:"
                      + servers[i], t);
      return results;
```

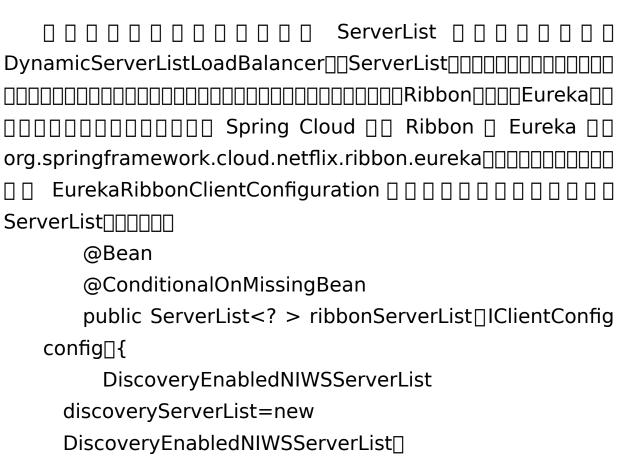
```
return null;
   } else {
    try {
      return rule.choose∏key∏;
    } catch∏Throwable t∏{
      return null:
    }
   }
  }
 BaseLoadBalancer
public void addServers[List<Server> newServers[]{
  if new Servers!=null && new Servers.size □□> 0 □ {
    try {
     ArrayList<Server>
                       newList=new
   ArrayList<Server>□□;
     newList.addAll∏allServerList∏;
     newList.addAll∏newServers∏;
     setServersList∏newList∏;
    } catch∏Exception e∏{
```

```
logger.error 

"Exception
                                       adding
                                while
     Servers",e□:
       }
    }
    }
   markServerDown[]Server server[][][][][][][][][][][]
    public void markServerDown∏Server server∏{
     if∏server==null∏{
       return;
     }
     if□! server.isAlive□□□{
       return;
     logger.error | "LoadBalancer: markServerDown called
   on["
          +server.getId[[]+"]"[];
     server.setAlive∏false∏;
     notifyServerStatusChangeListener
                                      singleton
                                 П
   □server□□;
   BaseLoadBalancer
    public List<Server> getReachableServers□□{
     return Collections.unmodifiableList∏upServerList∏;
    }
```

| ●getAllServers□□□□□□□□□□□□□□□BaseLoadBalancer□□□ |
|---|
| |
| public List <server> getAllServers□□{</server> |
| return Collections.unmodifiableList allServerList ; |
| } |
| DynamicServerListLoadBalancer |
| DynamicServerListLoadBalancer [][][] BaseLoadBalancer |
| |
| ServerList |
| <pre>DynamicServerListLoadBalancer []][][][][][][][][][][][][][][][][][][</pre> |
| } |
| |
| $getUpdatedListOfServers \verb $ |
| |





config[]; DomainExtractingServerList

serverList=new

DomainExtractingServerList[]

```
public class DomainExtractingServerList implements ServerList<DiscoveryEnabledServer> {
   private ServerList<DiscoveryEnabledServer> list;
   private IClientConfig clientConfig;
   private boolean approximateZoneFromHostname;
   public DomainExtractingServerList(ServerList<DiscoveryEnabledServer> list,
         IClientConfig clientConfig, boolean approximateZoneFromHostname) {
      this.list = list;
      this.clientConfig = clientConfig;
      this.approximateZoneFromHostname = approximateZoneFromHostname;
   @Override
   public List<DiscoveryEnabledServer> getInitialListOfServers() {
      List<DiscoveryEnabledServer> servers = setZones(this.list
            .getInitialListOfServers());
      return servers;
   @Override
   public List<DiscoveryEnabledServer> getUpdatedListOfServers() {
      List<DiscoveryEnabledServer> servers = setZones(this.list
            .getUpdatedListOfServers());
      return servers;
   }
```

| □□ DiscoveryEnabledNIWS | 6ServerList |
|---|--|
| |]obtainServersViaDiscovery |
| | |
| @Override | |
| public | List <discoveryenabledserver></discoveryenabledserver> |
| $getInitialListOfServers \hbox{\sqsubseteq} \{$ | |
| return obtainServer | sViaDiscovery∏∏; |
| } | |
| @Override | |
| public | List <discoveryenabledserver></discoveryenabledserver> |
| getUpdatedListOfServers□□ | { |
| return obtainServer | sViaDiscovery∏∏; |
| } | |
| □obtainServersViaDiscove | eryEurekaClient |
| | , |
| Instance | eInfo EurekaClient |
| Instance | eInfo[[[[]EurekaClient[[[][[][[][[][[][[] |
| | eInfo EurekaClient |

```
private List<DiscoveryEnabledServer> obtainServersViaDiscovery() {
       List<DiscoveryEnabledServer> serverList = new
ArrayList<DiscoveryEnabledServer>();
       if (eurekaClientProvider == null || eurekaClientProvider.get() == null) {
          logger.warn("EurekaClient has not been initialized yet, returning an empty
list");
          return new ArrayList<DiscoveryEnabledServer>();
       EurekaClient eurekaClient = eurekaClientProvider.get();
       if (vipAddresses!=null) {
          for (String vipAddress : vipAddresses.split(",")) {
             List<InstanceInfo> listOfInstanceInfo =
eurekaClient.getInstancesByVipAddress(
                      vipAddress, isSecure, targetRegion);
             for (InstanceInfo ii : listOfInstanceInfo) {
                if (ii.getStatus().equals(InstanceStatus.UP)) {
                    // 省略了一些实例信息的加工逻辑
                    DiscoveryEnabledServer des = new DiscoveryEnabledServer(ii,
isSecure, shouldUseIpAddr);
                   des.setZone(DiscoveryClient.getZone(ii));
                    serverList.add(des);
             if (serverList.size()>0 && prioritizeVipAddressBasedServers) {
                break;
       return serverList;
```

DiscoveryEnabledNIWSServerList DeurekaClient DeurekaClient

private List<DiscoveryEnabledServer> setZones

[]List<DiscoveryEnabledServer>

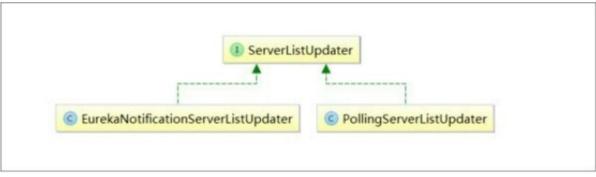
```
List<DiscoveryEnabledServer>
                                  result=new
    ArrayList<>∏;
      boolean
    isSecure=this.clientConfig.getPropertyAsBoolean[]
           CommonClientConfigKey.IsSecure,Boolean.TR
        UEΠ;
      boolean
    shouldUseIpAddr=this.clientConfig.getPropertyAsBoole
    an∏
           CommonClientConfigKey.UseIPAddrForServer,
        Boolean.FALSE∏;
      for

☐DiscoveryEnabledServer server: servers

☐{
        result.add
                         DomainExtractingServer
                 П
                    new
     ∏server,isSecure,shouldUselpAddr,
            this.approximateZoneFromHostname∏;
      }
      return result;
  ServerListUpdater
  protected
                    ServerListUpdater.UpdateAction
              final
 updateAction=new ServerListUpdater.UpdateAction□□{
     @Override
```

servers∏{

```
public void doUpdate□□{
    updateListOfServers□□;
   }
   };
  protected volatile ServerListUpdater serverListUpdater;
  public interface ServerListUpdater {
   public interface UpdateAction {
    void doUpdate□□;
   void start⊓UpdateAction updateAction∏;
   //0000000
   void stop□□;
   String getLastUpdate∏;
   long getDurationSinceLastUpdateMs□□;
   int getNumberMissedCycles□□;
   int getCoreThreads∏;
   }
```



| ● PollingServerListUpdater 🗆 🗎 🗎 🗎 🗎 🗎 🗎 🗎 |
|---|
| DynamicServerListLoadBalancer |
| |
| ■ EurekaNotificationServerListUpdater [] [] [] [] [] [] [] [] |
| DynamicServerListLoadBalancer 🛛 🗎 🗎 🗎 🗎 🗎 🗎 🗎 🗎 🗎 🗎 🗎 |
| PollingServerListUpdater |
| |
| PollingServerListUpdater |
| |
| Runnable |
| updateAction.doUpdateRunnable |
| |

```
@Override
public synchronized void start(final UpdateAction updateAction) {
   if (isActive.compareAndSet(false, true)) {
      final Runnable wrapperRunnable = new Runnable() {
         @Override
         public void run() {
            if (!isActive.get()) {
               if (scheduledFuture != null) {
                   scheduledFuture.cancel(true);
               return;
            }
            try {
               updateAction.doUpdate();
                lastUpdated = System.currentTimeMillis();
            } catch (Exception e) {
               logger.warn("Failed one update cycle", e);
      };
      scheduledFuture = getRefreshExecutor().scheduleWithFixedDelay(
            wrapperRunnable,
            initialDelayMs,
            refreshIntervalMs,
            TimeUnit.MILLISECONDS
      );
   } else {
      logger.info("Already active, no-op");
```

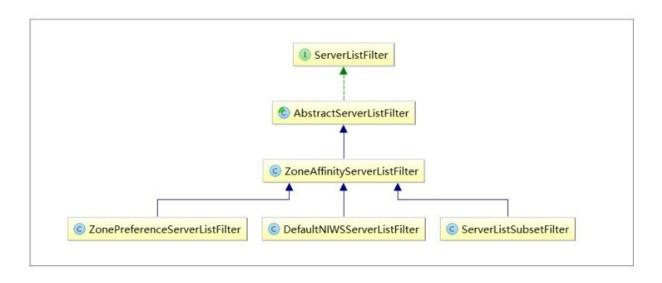
| PollingServerListUpdater |
|---|
| 🛮 🖟 🖟 🖟 🖟 InitialDelayMs 🖟 refreshIntervalMs 🖺 🗎 🗎 🗎 🗎 🗎 🗘 1000 🕻 |
| 30*1000 |
| 000000000000000000000000000000000000000 |
| ServerListUpdater |
| |

ServerListFilter

| updateAction.doUpdate | | | | | | [| |
|-----------------------|--|--|--|--|--|---|--|

```
updateListOfServers
    public void updateListOfServers□□{
      List<T> servers=new ArrayList<T><math>\sqcap \sqcap;
      if

| serverListImpl !=null
| {
       servers=serverListImpl.getUpdatedListOfServers
    \Pi\Pi;
       LOGGER.debug [ "List of Servers for {} obtained
    from Discovery client: {}",
          getIdentifier∏,servers∏;
       if∏filter !=null∏{
         servers=filter.getFilteredListOfServers
       ∏servers∏;
         LOGGER.debug [ "Filtered List of Servers for
       {} obtained from Discovery
   client: {}",
             getIdentifier∏, servers∏;
       }
      }
      updateAllServerList∏servers∏;
    }
  ServerListFilter□□□□
```



```
□□□□□ ZonePreferenceServerListFilter □□□□ Spring Cloud
 Ribbon | Netflix Ribbon
public
                                                                   abstract
                                                                                                          class
                                                                                                                                                         AbstractServerListFilter<T
         extends Server> implements ServerListFilter<T> {
                                            private volatile LoadBalancerStats stats;
                                            public
                                                                                                                void
                                                                                                                                                                              setLoadBalancerStats
                          □LoadBalancerStats stats□{
                                                    this.stats=stats:
                                            public LoadBalancerStats getLoadBalancerStats [[] {
                                                    return stats:
```

```
■ ZoneAffinityServerListFilter [] [] [] [] [] [] [] [] Zone
Affinity " " | TOTAL |
public List<T> getFilteredListOfServers [] List<T>
                    servers∏{
                                                 if | zone !=null && | zoneAffinity | | zoneExclusive | |
                              && servers !=null &&
                             servers.size∏> 0∏{
                                                           List<T>
                                                                                                                                    filteredServers=Lists.newArrayList
                                       ∏Iterables.filter∏
                                                                                        servers, this. zone Affinity Predicate. get Serve
                                                                     rOnlyPredicate \sqcap \sqcap \sqcap \sqcap;
                                                           if∏shouldEnableZoneAffinity∏filteredServers∏∏{
                                                                               return filteredServers:
                                                            } else if□zoneAffinity□{
                                                                               overrideCounter.increment∏;
                                                            }
                                                  return servers:
                   servers, this. zone Affinity Predicate. get Server Only Predicate
On the control of the
```

- - ●activeRequestsPerServer□□□□□□□ >=0.6□

```
if∏! zoneAffinity &&! zoneExclusive∏{
   return false:
 if∏zoneExclusive∏{
   return true;
 }
 LoadBalancerStats stats=getLoadBalancerStats□□;
 if∏stats==null∏{
   return zoneAffinity;
 } else {
   logger.debug [ "Determining if zone affinity
should
              enabled
                        with
                               given
                                                list:
        be
                                       server
{}",filtered∏;
   ZoneSnapshot snapshot=stats.getZoneSnapshot
```

ZoneSnapshot snapshot=stats.getZoneSnapshot [filtered[];

```
double
     loadPerServer=snapshot.getLoadPerServer□□;
               instanceCount=snapshot.getInstanceCount
         int
     \Box\Box;
         int
     circuitBreakerTrippedCount=snapshot.getCircuitTripp
     edCount∏∏;
         if
                                           double
                                  П
                                                         П
     circuitBreakerTrippedCount∏/instanceCount >=
   blackOutServerPercentageThreshold.get□□
                                            loadPerServer
           Ш
       >=activeRegeustsPerServerThreshold.get□□
           ||□instanceCount-circuitBreakerTrippedCount□<
   availableServersThreshold.get□□□{
           logger.debug
                                     "zoneAffinity
                                                        is
       overriden.blackOutServerPercentage: {},
   activeRegeustsPerServer: {},availableServers: {}",
                      Object[]{
                                     П
                                            double
                                                         new
       circuitBreakerTrippedCount /instanceCount,
   loadPerServer.instanceCount-
circuitBreakerTrippedCount}[];
           return false:
         } else {
           return true;
         }
     }
```

| ● DefaultNIWSServerListFilter 🛛 🗎 🗎 🗎 🗎 🗎 |
|--|
| $Zone Affinity Server List Filter \verb NIWS \verb Netflix Internal Web$ |
| Service DDDD |
| ServerListSubsetFilter |
| $Zone Affinity Server List Filter {\tt Community} {\tt Commun$ |
| $\\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ $ |
| |
| |
| 1.00``0000``000000000000000000000000000 |
| 2.000000000000000000000000000000000000 |
| |
| a0 |
| <cli>entName>.</cli> |
| <namespace>.ServerListSubsetFilter.eliminat</namespace> |
| $ionConnectionThresold \square$ |
| b |
| <cli>entName>.</cli> |
| <namespace>.ServerListSubsetFilter.eliminat</namespace> |
| ionFailureThresold[] |
| c. |
| □ 10% □ □ □ □ □ <clientname>.</clientname> |
| $< nameSpace > . ServerListSubsetFilter. forceEliminatePercent \ []$ |
| |
| 3.000000000000000000000000000000000000 |
| |
| <cli>clientName>.<namespace>.ServerListSubsetFilter.size[]</namespace></cli> |

```
OneAffinityServerListFilter
____Zone
   @Override
   public
          List<Server>
                     getFilteredListOfServers
 □List<Server> servers□{
     List<Server>
   output=super.getFilteredListOfServers[];
     if∏this.zone !=null && output.size∏==servers.size
   \Box\Box\Box
      List<Server> local=new ArrayList<Server>□□;
      for∏Server server : output∏{
         if
                   this.zone.equalsIgnoreCase
      local.add

| server

| ;
         }
      }
      if□! local.isEmpty□□□{
         return local;
      }
     return output;
```

ZoneAwareLoadBalancer

| ZoneAwareLoadBalancer | | | | | | | |
|---|--------|--------|--------|-------|------|------|-----|
| Dynamic Server List Load Balance | r | | | | | | |
| Dynamic Server List Load Balance | r 🔲 🛚 | | 10000 | | | | |
| chooseServer | Basel | _oad | Balar | icer[| | | |
| RoundRobinRule | | | | | | | |
| Zone Zone Zone | ne[] | | | | | | |
| Zone | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| 🛮 🗎 🖟 ZoneAwareLoadBalaı | ncer | | | | | | |
| setServersList | | | | | | | |
| setServerListForZones | | | | | | | |
| ☐Map <string,list<server>>zor</string,list<server> | neSe | rver | sMap | | | | |
| | | | | | | | |
| DynamicServerListLoadBalance | r[][] | | | | | | |
| public void setServersList | □List | : Isrv | /□{ | | | | |
| super.setServersList[] | _ | | | | | | |
| List <t> serverList=□</t> | List< | :T>[|]lsrv; | | | | |
| Map <string,list<serv< td=""><td>/er></td><td>></td><td>serve</td><td>ersIn</td><td>Zon</td><td>es=r</td><td>iew</td></string,list<serv<> | /er> | > | serve | ersIn | Zon | es=r | iew |
| HashMap <string,list<ser< td=""><td>ver></td><td>>>[[</td><td>];</td><td></td><td></td><td></td><td></td></string,list<ser<> | ver> | >>[[|]; | | | | |
| ••• | | | | | | | |
| setServerListForZones | s[]se | rver | sInZo | nes[] | ; | | |
| } | | | | | | | |
| protected void | | S | etSer | verL | istF | orZo | nes |
| <pre></pre> | zone | Serv | /ersM | ap∏{ | | | |
| LOGGER.debug 🛛 "Se | etting | j se | erver | list | for | zon | es: |
| {}",zoneServersMap□; | | | | | | | |

```
getLoadBalancerStats
    □□.updateZoneServerMapping□zoneServersMap□;
    }
  □□□□□□□□ LoadBalancerStats □□□□ ZoneStats □□□ Map
\squareZoneAwareLoadBalancer\square\squaresetServerListForZones\square\square\square\square
protected
                 void
                           setServerListForZones
 □Map<String,List<Server>> zoneServersMap□{
     super.setServerListForZones□zoneServersMap□;
     if∏balancers==null∏{
       balancers=new
    ConcurrentHashMap<String,BaseLoadBalancer>□□;
     }
             Map.Entry<String,List<Server>>
                                       entry:
  zoneServersMap.entrySet∏∏∏{
       String zone=entry.getKey[].toLowerCase[];
       getLoadBalancer
                                 .setServersList
                     П
                        zone
                             П
    ∏entry.getValue∏∏;
     }
                Map.Entry<String,BaseLoadBalancer>
     for
  existingLBEntry: balancers.entrySet∏∏∏{
```

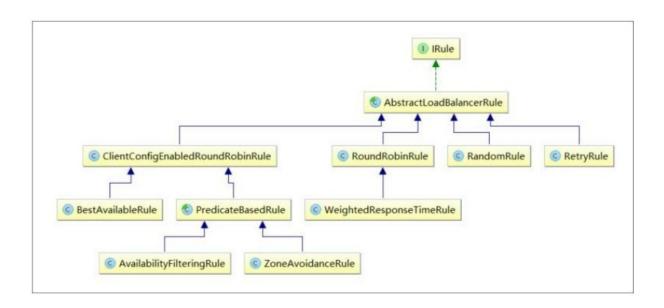
| if 🛘 ! zoneServersMap.keySet 🖟 🗘 .contains |
|---|
| <pre> existingLBEntry.getKey {</pre> |
| existingLBEntry.getValue 🛮 🗘 .setServersList |
| <pre>□Collections.emptyList□□□;</pre> |
| } |
| } |
| } |
| |
| 0000000Zone00000000000000000000000000000 |
| getLoadBalancer |
| ☐ IRule ☐☐☐☐☐☐☐ AvailabilityFilteringRule☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| ZoneZoneZone |
| balancers DDDZone DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Zone |
| |
| |

```
public Server chooseServer(Object key) {
       if (!ENABLED.get() || getLoadBalancerStats().getAvailableZones().size() <= 1) {</pre>
          logger.debug("Zone aware logic disabled or there is only one zone");
          return super.chooseServer(key);
       Server server = null;
       try {
          LoadBalancerStats lbStats = getLoadBalancerStats();
          Map<String, ZoneSnapshot> zoneSnapshot = ZoneAvoidanceRule.createSnapshot
(lbStats);
          logger.debug("Zone snapshots: {}", zoneSnapshot);
          Set<String> availableZones = ZoneAvoidanceRule.getAvailableZones
(zoneSnapshot, triggeringLoad.get(), triggeringBlackoutPercentage.get());
          logger.debug("Available zones: {}", availableZones);
          if (availableZones != null && availableZones.size() <
zoneSnapshot.keySet().size()) {
             String zone = ZoneAvoidanceRule.randomChooseZone(zoneSnapshot,
availableZones);
             logger.debug("Zone chosen: {}", zone);
             if (zone != null) {
                 BaseLoadBalancer zoneLoadBalancer = getLoadBalancer(zone);
                 server = zoneLoadBalancer.chooseServer(key);
       } catch (Throwable e) {
          logger.error("Unexpected exception when choosing server using zone aware
logic", e);
       if (server != null) {
          return server;
          logger.debug("Zone avoidance logic is not invoked.");
          return super.chooseServer(key);
```



П

| $\verb , triggering Blackout Percentage.get $ |
|---|
| 000000Zone000000000000000000000000000000 |
| ■ |
| |
| ■ |
| □Zone□□□ |
| ■0000000000000000000000000000000000000 |
| Zone Zone Zone |
| |
| ●□□□□□Zone□□□□□□□□□□□□□Zone□□□□□□□□□□□Zone□ |
| |
| ● |
| chooseServer |
| choose |
| ZoneAvoidanceRule 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 |
| ZoneAvoidanceRule |
| |
| |
| |
| Ribbon |
| |
| Ribbon |
| |
| Ribbon |
| $RoundRobinRule \verb Zone Avoidance Rule \verb \verb \verb \verb \verb \verb $ |
| |



AbstractLoadBalancerRule

RandomRule

```
@Override
public Server choose (Object key) {
   return choose (getLoadBalancer(), key);
public Server choose(ILoadBalancer lb, Object key) {
   Server server = null;
   while (server == null) {
      if (Thread.interrupted()) {
        return null;
      List<Server> upList = lb.getReachableServers();
      List<Server> allList = lb.getAllServers();
      int serverCount = allList.size();
      if (serverCount == 0) {
         return null;
      int index = rand.nextInt(serverCount);
      server = upList.get(index);
      if (server == null) {
         Thread.yield();
         continue;
      if (server.isAlive()) {
        return (server);
      server = null;
      Thread.yield();
   return server;
```

RoundRobinRule

RandomRule Count C

```
public Server choose (ILoadBalancer lb, Object key) {
   Server server = null;
   int count = 0;
   while (server == null && count++ < 10) {
      List<Server> reachableServers = lb.getReachableServers();
     List<Server> allServers = lb.getAllServers();
      int upCount = reachableServers.size();
      int serverCount = allServers.size();
      if ((upCount == 0) || (serverCount == 0)) {
         log.warn("No up servers available from load balancer: " + lb);
         return null;
      int nextServerIndex = incrementAndGetModulo(serverCount);
      server = allServers.get(nextServerIndex);
      if (server == null) {
         Thread.yield();
        continue;
      if (server.isAlive() && (server.isReadyToServe())) {
         return (server);
      server = null;
   if (count >= 10) {
      log.warn("No available alive servers after 10 tries from load balancer: "
            + lb);
   return server;
```

RetryRule

```
null∏
    public
                         RetryRule
               class
                                       extends
 AbstractLoadBalancerRule {
     IRule subRule=new RoundRobinRule∏;
      long maxRetryMillis=500;
      public Server choose∏LoadBalancer lb,Object key∏{
       long requestTime=System.currentTimeMillis□□;
       long deadline=requestTime+maxRetryMillis;
       Server answer=null:
       answer=subRule.choose∏key∏;
       if□□□answer==null□||□! answer.isAlive□□□□
          InterruptTask task=new InterruptTask deadline
           -System.currentTimeMillis∏∏;
        while \square! Thread.interrupted \square
          answer=subRule.choose[key[];
          if \square \square \exists nswer = = null \square | | \square | answer.isAlive \square \square \square
                ☐ System.currentTimeMillis
          deadline∏∏{
           Thread.yield\square;
          } else {
           break:
```

```
}
                                                                                      task.cancel∏∏;
                                                                         }
                                                                         if = answer = null = answer.isAlive = 
                                                                                      return null;
                                                                         } else {
                                                                                      return answer;
                                                                         }
                                                           }
                             WeightedResponseTimeRule
                             One of the control of
WeightedResponseTimeRule | | | | | | | | | | | | | | | |
serverWeightTimer.schedule new DynamicServerWeightTask
class DynamicServerWeightTask extends TimerTask {
                                                           public void run□□{
                                                                         ServerWeight serverWeight=new ServerWeight□□;
                                                                        try {
                                                                                      serverWeight.maintainWeights∏;
                                                                         } catch∏Throwable t∏{
```

```
logger.error [] "Throwable caught while running
      DynamicServerWeightTask for"+name,t□;
       }
      }
    }
   ______List<Double>
accumulatedWeights=new ArrayList<Double> | | | | List | | | | |
public void maintainWeights∏∏{
     ILoadBalancer lb=getLoadBalancer□□;
     try {
       logger.info∏"Weight adjusting job started"∏;
       AbstractLoadBalancer
                                          nlb =
    ∏AbstractLoadBalancer∏lb;
       LoadBalancerStats stats=nlb.getLoadBalancerStats
    \Pi\Pi;
       //____totalResponseTime
       double totalResponseTime=0;
       for Server server : nlb.getAllServers [] [] {
        ss=stats.getSingleServerStat
        ServerStats
     ∏server∏;
        totalResponseTime+=ss.getResponseTimeAvg∏;
```

```
Double weightSoFar=0.0;
      List<Double>
                           finalWeights=new
   ArrayList<Double>□□;
      for Server server : nlb.getAllServers □□□{
                    ss=stats.getSingleServerStat
       ServerStats
    ∏server∏;
       double
                    weight=totalResponseTime-
     ss.getResponseTimeAvg∏∏;
       weightSoFar+=weight;
       finalWeights.add\u00dd\u00ddveightSoFar\u00dd;
      }
      setWeights∏finalWeights∏;
     } catch∏Throwable t∏{
      logger.error 
☐ "Exception
                              dynamically
                        while
   calculating server weights",t[];
     } finally {
      serverWeightAssignmentInProgress.set∏false∏;
     }
  ullet
```

}

| weightSoFar |
|--------------------------------------|
| |
| |
| 40 80 100 10+40+80+100=230 |
| |
| ●□□A:230-10=220 |
| ●□□B:220+□230-40□=410 |
| ●□□C:410+□230-80□=560 |
| ●□□D:560+□230-100□=690 |
| |
| |
| |
| accumulatedWeights |
| |
| ●□□A:[0,220] |
| ●□□B:□220,410] |
| ●□□C:□410,560] |
| ●□□D:□560,690□ |
| |
| |
| |
| |
| WeightedResponseTimeRule |
| |
| |

```
public Server choose (ILoadBalancer lb, Object key) {
         List<Double> currentWeights = accumulatedWeights;
         List<Server> allList = lb.getAllServers();
         int serverCount = allList.size();
         if (serverCount == 0) {
           return null;
         }
         int serverIndex = 0;
         // 获取最后一个实例的权重
         double maxTotalWeight = currentWeights.size() == 0 ? 0 : currentWeights.get
(currentWeights.size() - 1);
         if (maxTotalWeight < 0.001d) {
            // 如果最后一个实例的权重值小于 0.001, 则采用父类实现的线性轮询的策略
            server = super.choose(getLoadBalancer(), key);
            if(server == null) {
               return server;
         } else {
            // 如果最后一个实例的权重值大于等于 0.001, 就产生一个 [0, maxTotalWeight) 的随机数
            double randomWeight = random.nextDouble() * maxTotalWeight;
            int n = 0;
            for (Double d : currentWeights) {
            // 遍历维护的权重清单, 若权重大于等于随机得到的数值, 就选择这个实例
               if (d >= randomWeight) {
                  serverIndex = n;
                 break;
               } else {
                 n++;
            server = allList.get(serverIndex);
         }
      return server;
```

```
ClientConfigEnabledRoundRobinRule
 RoundRobinRule
ClientConfigEnabledRoundRobinRule
  public
      class
extends AbstractLoadBalancerRule {
    RoundRobinRule
                 roundRobinRule=new
  RoundRobinRule∏;
    @Override
    public Server choose \ Object key \ {
     if∏roundRobinRule !=null∏{
      return roundRobinRule.choose∏key∏;
     } else {
      throw new IllegalArgumentException

☐
         "This class has not been initialized with
       the RoundRobinRule class"∏;
     }
    }
```

}

BestAvailableRule

```
public Server choose (Object key) {
   if (loadBalancerStats == null) {
     return super.choose(key);
   List<Server> serverList = getLoadBalancer().getAllServers();
   int minimalConcurrentConnections = Integer.MAX VALUE;
   long currentTime = System.currentTimeMillis();
   Server chosen = null;
   for (Server server: serverList) {
      ServerStats serverStats = loadBalancerStats.getSingleServerStat(server);
      if (!serverStats.isCircuitBreakerTripped(currentTime)) {
         int concurrentConnections = serverStats.getActiveRequestsCount(currentTime);
         if (concurrentConnections < minimalConcurrentConnections) {</pre>
            minimalConcurrentConnections = concurrentConnections;
            chosen = server;
   if (chosen == null) {
      return super.choose(key);
   } else {
      return chosen;
```

PredicateBasedRule

| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
|--|
| |
| |
| AbstractServerPredicate choose |
| $AbstractServer Predicate \verb choose Round Robin After Filtering \verb \verb $ |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| public abstract class PredicateBasedRule extends |
| ClientConfigEnabledRoundRobinRule { |
| public abstract AbstractServerPredicate |
| getPredicate□□; |
| @Override |
| public Server choose Object key { |
| <pre>ILoadBalancer lb=getLoadBalancer□□;</pre> |
| Optional <server> server=getPredicate</server> |
| .chooseRoundRobinAfterFiltering |
| <pre> Ib.getAllServers ,key ;</pre> |
| if[]server.isPresent[][][]{ |
| return server.get□□; |
| } else { |
| return null; |
| } |
| } |
| } |
| AbstractServerPredicate |
| choose [] [] [] choose Round Robin After Filtering [] [] [] [] [] |

```
Optional.absent
      public
                         class
                                AbstractServerPredicate
               abstract
   implements Predicate<PredicateKey> {
      public
                                     Optional < Server >
   chooseRoundRobinAfterFiltering
                                   П
                                         List<Server>
   servers,Object loadBalancerKey[]{
                             eligible=getEligibleServers
          List<Server>
      ∏servers,loadBalancerKey∏;
          if∏eligible.size∏==0∏{
            return Optional.absent∏;
          }
                     Optional.of
                                           eligible.get
          return
      □nextIndex.getAndIncrement□□% eligible.
     size□□□□;
        }
        public
                                     getEligibleServers
                    List<Server>
     □List<Server> servers,Object loadBalancerKey□{
          if∏loadBalancerKey==null∏{
            return ImmutableList.copyOf [] Iterables.filter
        □servers,this.getServerOnlyPredicate□□□□;
          } else {
            List<Server> results=Lists.newArrayList□□;
            for∏Server server: servers∏{
             if
                    this.apply
                                         PredicateKey
                               П
                                   new
          □loadBalancerKey,server□□□{
```

```
results.add∏server∏;
                                                                   }
                                                           }
                                                           return results:
                                                  }
                                          }
                 DODO TO THE STATE OF THE STATE 
apply \( \text{AbstractServerPredicate} \) \( \partial \text{The product of the p
                                                             AbstractServerPredicate
П
                                                                                                                                                                                               П
                                                                                                                                                                                                                              П
com.google.common.base.Predicate
Google Guava
                                                                                          Collections □ □ □ Java Collections
 Framework
```

AvailabilityFilteringRule

```
____PredicateBasedRule
DDD"DDDDDDDDDDDDDDAvailabilityPredicateD
    public
                      AvailabilityPredicate extends
             class
 AbstractServerPredicate {
       public boolean apply □ @Nullable PredicateKey
    input∏{
         LoadBalancerStats stats=getLBStats[];
         if∏stats==null∏{
            return true;
         }
                                   shouldSkipServer
         return
      □stats.getSingleServerStat□input.getServer□□□□;
       private boolean shouldSkipServer ☐ ServerStats
    stats∏{
         if □ □ CIRCUIT BREAKER FILTERING.get □ □ &&
      stats.isCircuitBreakerTripped
                  stats.getActiveRequestsCount
                                               П
           return true;
         return false;
       }
    }
   _____shouldSkipServer_____
```

```
ullet
 <nameSpace>.ActiveConnectionsLimit
 _____apply ___ false_______
\Pi\Pi\Pi\Pi\Pitrue\Pi
 public Server choose∏Object key∏{
   int count=0:
   Server server=roundRobinRule.choose⊓key□;
   while \mid count + + < = 10 \mid {
    return server;
    }
    server=roundRobinRule.choose[]key[];
   }
   return super.choose∏key∏;
 ZoneAvoidanceRule
```

| Zone createSnapshot |
|--|
| getAvailableZones |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| CompositePredicate |
| ZoneAvoidancePredicate |
| |
| public class ZoneAvoidanceRule extends |
| PredicateBasedRule { |
| |
| private CompositePredicate compositePredicate; |
| public ZoneAvoidanceRule□□{ |
| super□□; |
| ZoneAvoidancePredicate zonePredicate=new |
| ZoneAvoidancePredicate[]this[]; |
| AvailabilityPredicate availabilityPredicate=new |
| AvailabilityPredicate[]this[]; |
| compositePredicate=createCompositePredicate |
| □zonePredicate, |
| availabilityPredicate[]; |
| } |
| *** |
| } |
| ZoneAvoidanceRule |
| |
| ZoneAvoidancePredicate |
| AvailabilityPredicate CompositePredicate |
| CompositePredicate [] [] [] [] [] [] [] [] [] [|

```
public class CompositePredicate extends AbstractServerPredicate {
       private AbstractServerPredicate delegate;
       private List<AbstractServerPredicate> fallbacks = Lists.newArrayList();
       private int minimalFilteredServers = 1;
       private float minimalFilteredPercentage = 0;
       @Override
       public List<Server> getEligibleServers(List<Server> servers, Object
loadBalancerKey) {
          List<Server> result = super.getEligibleServers (servers, loadBalancerKey);
          Iterator<AbstractServerPredicate> i = fallbacks.iterator();
          while (!(result.size() >= minimalFilteredServers && result.size() > (int)
(servers.size() * minimalFilteredPercentage))
                 && i.hasNext()) {
             AbstractServerPredicate predicate = i.next();
             result = predicate.getEligibleServers(servers, loadBalancerKey);
          }
          return result;
```

____getEligibleServers

- ullet
- ullet

 $0 \square \square$



| Ribbon |
|--|
| |
| Spring Cloud Ribbon |
| Ribbon |
| IClientConfig:Ribbon |
| $com.netflix.client.config.DefaultClientConfigImpl \verb $ |
| ● IRule:Ribbon |
| com.netflix.loadbalancer.ZoneAvoidanceRule |
| |
| ● IPing:Ribbon |
| com.netflix.loadbalancer.NoOpPing |
| 0000000000000000true |
| • ServerList <server> [</server> |
| com.netflix.loadbalancer.ConfigurationBasedServerList |
| ● ServerListFilter <server> □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □</server> |
| org.springframework.cloud.netflix.ribbon.ZonePreferenceSer |
| verLis tFilter |
| ILoadBalancer |
| com.netflix.loadbalancer.ZoneAwareLoadBalancer |
| |

| Spring Cloud Eureka |
|---|
| Eureka_Ribbon |
| |
| Spring Boot |
| NoOpPing |
| @Configuration |
| <pre>public class MyRibbonConfiguration {</pre> |
| @Bean |
| public IPing ribbonPing□IClientConfig config□{ |
| return new PingUrl□□; |
| } |
| } |
| @RibbonClient |
| hello-service |
| @Configuration |
| @RibbonClient name="hello- |
| $service", configuration = HelloServiceConfiguration.class \square$ |
| <pre>public class RibbonConfiguration {</pre> |
| } |
| |
| <u>Camden□□□RibbonClient□□□□</u> |
| |
| |
| Bean |
| |
| RibbonClient |

```
□□□□□□Camden□□□□Spring Cloud Ribbon□ RibbonClient □□□□
_____clientName>.ribbon.<key>=
<value>nnnnnnnnnnnnnnnnnnnnhhello-servicennnnn
hello-
 service.ribbon.NFLoadBalancerPingClassName=com.netflix
  .loadbalancer.PingUrl
   □□ hello-service □□□□□NFLoadBalancerPingClassName □□
□□□□□□□ IPing □□□□□□ Camden □□□□Spring Cloud Ribbon □□
                           П
П
                                                       П
org.springframework.cloud.netflix.ribbon.PropertiesFactory □□
\sqcap\sqcap\sqcap\sqcapRibbonClient\sqcap\sqcap\sqcap\sqcap\sqcap\sqcap\sqcap\sqcap
     public class PropertiesFactory {
       @Autowired
       private Environment environment;
                Map<Class,String>
                                     classToProperty=new
       private
   HashMap <> \square \square;
       public PropertiesFactory∏∏{
         classToProperty.put
     □ILoadBalancer.class, "NFLoadBalancerClassName" □;
         classToProperty.put
     □IPing.class, "NFLoadBalancerPingClassName"□;
         classToProperty.put
     □IRule.class,"NFLoadBalancerRuleClassName"□;
         classToProperty.put
     □ServerList.class,"NIWSServerListClassName"□;
```

```
classToProperty.put
 ☐ServerListFilter.class,"NIWSServerListFilterClassName"
 Π;
   }
   public boolean isSet∏Class clazz,String name∏{

  □ getClassName

              StringUtils.hasText
     return
 \sqcapclazz,name\sqcap \sqcap;
   }
   public String getClassName ☐ Class clazz,String
name∏{
     if | this.class To Property.contains Key | clazz | | {
       String
   classNameProperty=this.classToProperty.get[]clazz[];
                  className=environment.getProperty
       String
   return className:
     }
     return null;
   }
   @SuppressWarnings□"unchecked"□
   public <C> C get ☐ Class<C> clazz,IClientConfig
config,String name∏{
     String className=getClassName□clazz,name□;
     if□StringUtils.hasText□className□□{
       try {
         Class<?
                          toInstantiate=Class.forName
                    >
     ∏className∏;
```

```
instantiateWithConfig
        return
              П
                 C
                    П
      □toInstantiate,config□;
       } catch∏ClassNotFoundException e∏{
        throw
               new
                     IllegalArgumentException
      □"Unknown class to load"+className+" for class
      "+clazz+" named "+name∏;
      }
      return null;
    }
  ullet NFLoadBalancerPingClassName ldot ldot IPing ldot ldot ldot
  ●NIWSServerListClassName□□□ServerList□□□□□□
  ■NIWSServerListFilterClassName
□□□ServerListFilter□□□□
____Camden________RibbonClient__
ILoadBalancer | IPing | IRule | ServerList | ServerListFilter | | | | | |
```

| ribbon.ConnectTimeout=250 |
|--|
| 00000000000000000000000000000000000000 |
| ●□□□□□□□□□□ <client>.ribbon.<key>=<value>□□□□□□</value></key></client> |
|
key>_ <value><cli>ent></cli></value> |
| $\verb $ |
| $\verb $ |
| _ /hello restTemplate.getForEntity |
| $\verb "http://helloservice/hello", String.class .getBody $ |
| |
| hello- |
| service.ribbon.listOf Servers = local host: 8001, local host: 8002 |
| ,localhost:8003 |
| |
| $com.netflix.client.config.CommonClientConfigKey \verb $ |
| |
| |
| <u> </u> |
| <pre> [Spring Cloud []] [Spring Cloud Ribbon Spring </pre> |
| Cloud Eureka |
| ServerList П П П П П П П |
| com.netflix.niws.loadbalancer.DiscoveryEnabledNIWSServerL |
| ist DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| com.netflix.niws.loadbalancer.NIWSDiscoveryPing |
| |
| |

| | | | | Spring | Cloud | Eureka | | | | |
|-----|-----------|---------------|-----------------|------------|--------------|-------------|---|----------|----------|-------------------|
| org | .spri | ngfr | ame | work.clou | ıd.netflix.ı | ribbon.eure | eka.D | oma | inEx | ĸtr |
| act | ingSe | erve | rList | | | | | | | |
| | |][] AV | VS AI | MI | | | | | | |
| | □□S | pring | g Clo | ud Eurek | a | | | | | |
| | rello- | serv | ice.r | ibbon.list | :OfServer | 5 000000 | | | | |
| Eur | eka[| | | | □□□□□Rib | bon[][][[| | | | |
| | | | | | Eure | ka | <cli< td=""><td>ent></td><td></td><td></td></cli<> | ent> | | |
| | | | | | | | | | | |
| | | | Sprin | g Cloud I | Ribbon□□ | 0000000 | | | | |
| Eur | eka[| | | | | | | | | |
| | | | | 10000000 | | |]z | zone | | |
| | | | | | | | | | | |
| | eı | ureka | a.ins | tance.me | etadataMa | p.zone=sh | nangl | nai | | |
| | ∏Sp | ring | Clou | d Ribbon | □Spring C | Cloud Eurel | ka∏∏[| | | |
| | | |] <u> </u> Ει | ıreka∏Rik | bon[[[[| | | | | |
| | | | |][[[[[]]]] | client>.r | ibbon.listO | fServ | ers | | |
| | | | | | | | | | | |
| | rik | obor | ı.eur | eka.enab | led=false | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | | | | | 'UUUL | | _ | |
| |]∐∐∠(
 | 00Ke | epe | CPL | 100000000 | | | ⊔⊔∐E | urel | <a∐< td=""></a∐<> |
| | J∐∐[[| | | | JUUU0000 | | J∐∐∏[
 | 」∐"∐
 | ∐"[[
 | |
| | | | | | | | | AP 🛚 | | |

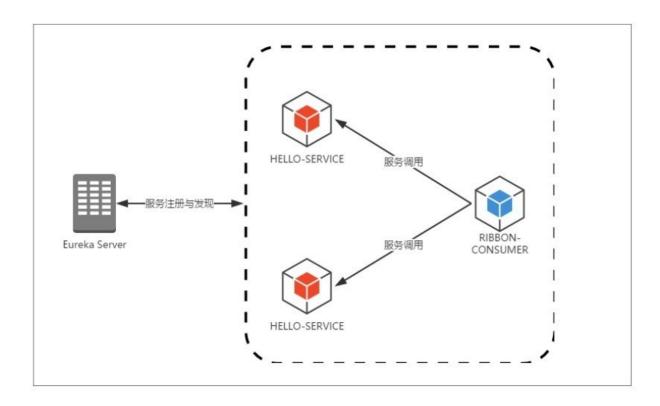
```
OSpring Cloud Eureka
□□□□□□Camden SR2□□□□□Spring Cloud□□□Spring Retry□□□
spring.cloud.loadbalancer.retry.enabled=true
  hystrix.command.default.execution.isolation.thread.tim
 eoutInMilliseconds=10000
  hello-service.ribbon.ConnectTimeout=250
  hello-service.ribbon.ReadTimeout=1000
  hello-service.ribbon.OkToRetryOnAllOperations=true
  hello-service.ribbon.MaxAutoRetriesNextServer=2
  hello-service.ribbon.MaxAutoRetries=1
 @ConfigurationProperties
□"spring.cloud.loadbalancer.retry"□
  public class LoadBalancerRetryProperties {
   private boolean enabled=false;
   }
```

| hystrix.command.default.execution.isolation.thread.tim |
|--|
| eoutIn Milliseconds |
| ◆hello-service.ribbon.ConnectTimeout□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| ●hello-service.ribbon.ReadTimeout□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| ullet hello-service.ribbon.OkToRetryOnAllOperations $lacktriangle$ |
| |
| ●hello-service.ribbon.MaxAutoRetriesNextServer |
| |
| ●hello-service.ribbon.MaxAutoRetries□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| MaxAutoRetries |
| MaxAutoRetriesNextServer |

<u>| 5| | | | | | | | Spring Cloud</u> <u>Hystrix</u>

| 000000000000000000000000000000000000000 |
|---|
| |





●eureka-server□□□□□□□□□□□□1111□

```
8081 8082
   ●ribbon-consume
   GET \Pi \Pi \Pi
http://localhost:9000/ribbon-consumer
    {
       "timestamp": 1473992080343,
       "status": 500,
       "error": "Internal Server Error",
       "exception":
    "org.springframework.web.client.ResourceAccessExcep
    tion",
       "message": "I/O
                       error on
                                 GET
                                      request for
    \"http://HELLO-SERVICE/hello\":Connection
                                          refused:
                               exception
    connect;
                   nested
    java.net.ConnectException:
                             Connection
                                         refused:
    connect",
       "path": "/ribbon-consumer"
    }
   □□□□□□□Spring Cloud Hystrix□
   ■□ ribbon-consumer □□□ pom.xml □ dependency □□□□□
springcloud-starter-hystrix□□□
    <dependency>
       <groupId>org.springframework.cloud</groupId>
       <artifactId>spring-cloud-starter-
    hystrix</artifactId>
    </dependency>
```

| ■ ribbon-consumer ConsumerApplication |
|---|
| @EnableCircuitBreaker |
| @EnableCircuitBreaker |
| @EnableDiscoveryClient |
| @SpringBootApplication |
| public class ConsumerApplication { |
| @Bean |
| @LoadBalanced |
| RestTemplate restTemplate [[] { |
| return new RestTemplate□□; |
| } |
| public static void main String[]args { |
| SpringApplication.run |
| <pre>ConsumerApplication.class,args[];</pre> |
| } |
| } |
| _ |
| @SpringCloudApplication |
| Spring Cloud |
| |
| @Target[]{ElementType.TYPE}[] |
| @Retention Retention Policy. RUNTIME |
| @Documented |
| @Inherited |
| @SpringBootApplication |
| @EnableDiscoveryClient |
| @EnableCircuitBreaker |

```
public @interface SpringCloudApplication {
   ConsumerController RestTemplate RestTemplate RestTemplate RestTemplate
@Service
      public class HelloService {
       @Autowired
       RestTemplate restTemplate;
       @HystrixCommand
    ∏fallbackMethod="helloFallback"∏
       public String helloService∏{
         return restTemplate.getForEntity 

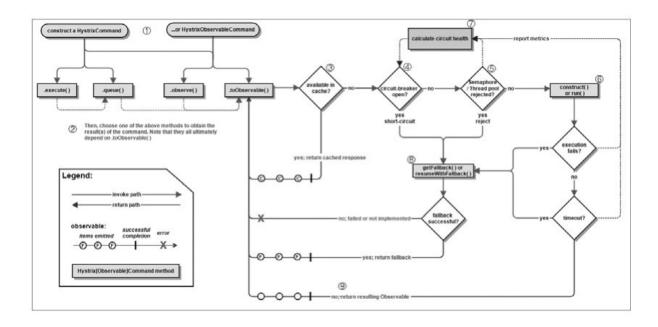
| "http://HELLO-
      SERVICE/hello",
    String.class∏.getBody∏∏;
       public String helloFallback∏∏{
         return "error";
       }
   helloConsumer□□□□□□
    @RestController
    public class ConsumerController {
       @Autowired
       HelloService helloService:
```

```
@RequestMapping
                                                                                                                                                                                                                   value="/ribbon-
                                                                                                                                                                         П
                           consumer",method=RequestMethod.GET[]
                                              public String helloConsumer∏∏{
                                                       return helloService.helloService[];
                                               }
                            }
                   Service RIBBONCONSUMER RIBBONCONSUME
SERVICE DO DO DO DO DO DO DO DO DE LA CASA DEL CASA DE LA CASA DE 
http://localhost:9000/ribbon-consumer
error
_/hello_____
                            @RequestMapping
         □value="/hello",method=RequestMethod.GET□
                           public String hello∏∏throws Exception {
                                               ServiceInstance
                           instance=client.getLocalServiceInstance□□;
                                              int sleepTime=new Random□□.nextInt□3000□;
                                              logger.info\\"sleepTime:\"+sleepTime\\;
                                              Thread.sleep∏sleepTime∏;
                                              logger.info | "/hello,host:"+instance.getHost | |
                            +",service id:"+instance.getServiceId∏∏;
                                              return "Hello World":
                            }
```

```
On Thread.sleep
@HystrixCommand
       ☐ fallbackMethod="helloFallback",commandKey="helloKey"
       П
                     public String hello□□{
                                    long start=System.currentTimeMillis∏;
                                    long end=System.currentTimeMillis∏;
                                    logger.info□"Spend time: "+□end-start□□;
                                    return result.toString∏;
              □□□□ HELLO-SERVICE □ RIBBON-CONSUMER □□□□□□□□□
http://localhost:9000/ribbon-consumer
CONSUMER CON
```



____Netflix Hystrix_____



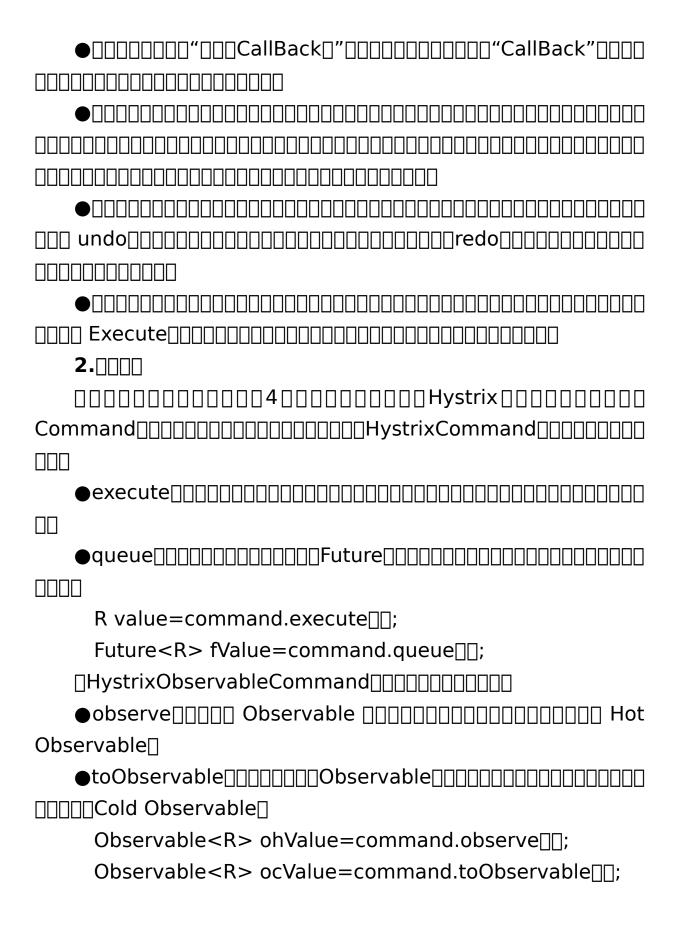
1. [] HystrixCommand [HystrixObservableCommand

//□□□ public class Receiver { public void action□□{

```
}
   }
  interface Command {
     void execute∏;
   }
  public class ConcreteCommand implements Command
{
     private Receiver receiver;
     this.receiver=receiver;
     public void execute∏∏{
       this.receiver.action□□;
     }
  public class Invoker {
     private Command command;
    public void setCommand ☐Command ☐{
     this.command=command;
    public void action∏∏{
       this.command.execute∏;
    }
```

```
public class Client {
                   public static void main String[]args {
                      Receiver receiver=new Receiver□□;
                                              command=new ConcreteCommand
                       Command
               ∏receiver∏;
                      Invoker invoker=new Invoker□□;
                      invoker.setCommand[]command[];
                      invoker.action∏; //∏∏∏∏∏∏∏∏∏∏∏
               }
       ullet Concrete Command lacksquare l
HystrixCommand []HystrixObservableCommand [][]Hystrix[]
```

}



```
RxJava_____
  00"0000"00000Subscriber0000000"000"000"000"000000
onNext□□□□□
  ● □ □ □ Observable □ □ □ □ □ □ □ □ □ □ Subscriber.on Completed
//ПППППobservable
  Observable < String > observable = Observable.create
□new Observable.OnSubscribe<String>□□{
    @Override
    public void call ☐ Subscriber<? super String>
  subscriber∏{
     subscriber.onNext∏"Hello RxJava"∏;
     subscriber.onNext□"I am □□□DD"□;
     subscriber.onCompleted∏;
    }
   }∏;
  Subscriber<String>
                      subscriber=new
 Subscriber<String>□□{
```

```
@Override
                           public void onCompleted□□{
                           }
                           @Override
                           public void onError Throwable e {\}
                           @Override
                           public void onNext□String s□{
                                System.out.println∏"Subscriber : "+s∏;
                           }
                };
                //\square\square
                observable.subscribe∏subscriber∏;
           subscriber no observable. subscribe subscriber no no subscriber no subscriber no subscriber no subscriber no no sub
           ______observable _____Hot Observable _
Cold Observable ☐ ☐ ☐ ☐ ☐ ☐ ☐ Command.observe ☐ ☐ ☐
command.toObservable
n"nnn"nnnnnnnnnnnnhhot Observablennn"n
חחחחח"חחחחחחחחחחחחחחחחחחחחחCold Observable
RxJava \verb||||||||execute||||||queue||||||||RxJava||||||||||||
●execute[][][][]queue[][][][][]Future<R>[]get[][][][][][]
```

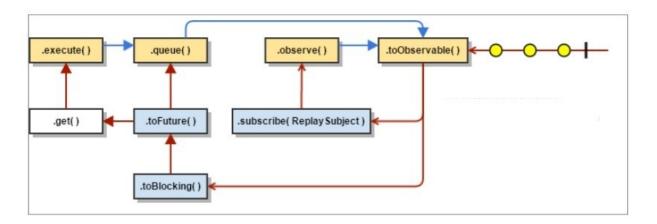
```
●queue□□□□□ toObservable□□□□□□ Cold Observable□□
_____Future__Observable______
public R execute □ [ {
     try {
      return queue□□.get□□;
     } catch∏Exception e∏{
      throw decomposeException □e□;
     }
   public Future<R> queue∏∏{
     final Observable\langle R \rangle o=toObservable|| \Box ||;
     final Future<R> f=o.toBlocking\square.toFuture\square;
     if∏f.isDone∏∏{
      }
     return f;
  3.000000
  Observable∏∏∏∏∏∏
  4.0000000
```

| ●□□□□□□□□□□Hystrix□□□□□□□□□□fallback□□□□□□□□□ |
|---|
| 8 |
| ●□□□□□□□□□□Hystrix□□□5□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| 5. 000/0000/00000 |
| Hystrix_ |
| fallback8 |
| Hystrix |
| Hystrix 🗆 🗆 🗆 🖂 🖺 Bulkhead |
| Pattern |
| 6.HystrixObservableCommand.construct |
| HystrixCommand.run 🗆 |
| Hystrix |
| ●HystrixCommand.run□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| ■ HystrixObservableCommand.construct □ □ □ □ □ □ |
| Observable |
| |
| D TimeoutException |
| Hystrixfallback8 |
| |
| 00000000000000000000000000000000000000 |
| run Hystrix Observable |
| onCompleted |
| Observable |
| 7. 0000000 |

| Hystrix"_0"_0"_0"_0"_0"_0"_0"_00"_000000000 |
|--|
| |
| 000000000000000000000000000000000000000 |
| |
| 8.fallback□□ |
| Hystrixfallback |
| |
| ●04000000"00/00"00000000000 |
| ●0500000000000000000000000000000000000 |
| ● 🛮 6 🗎 🗎 HystrixObservableCommand.construct 🖺 🖺 |
| HystrixCommand.run |
| |
| |
| $Hystrix Command \verb Hystrix Observable Command \verb $ |
| |
| <pre> HystrixCommand HystrixObservableCommand </pre> |
| |
| |
| HystrixCommand.getFallback |
| ● 🛮 🗘 🗎 HystrixObservableCommand 🗘 🗘 🗘 🗘 |
| $HystrixObservableCommand.resumeWithFallback \verb $ |
| Observable |
| Hystrix |
| $Hystrix Command.get Fallback \verb $ |
| <pre>getFallback </pre> |
| $HystrixObservableCommand.resumeWithFallback \verb $ |
| □Observable□□□□□□ |

- ●execute∏∏∏∏∏∏∏

- - 9.



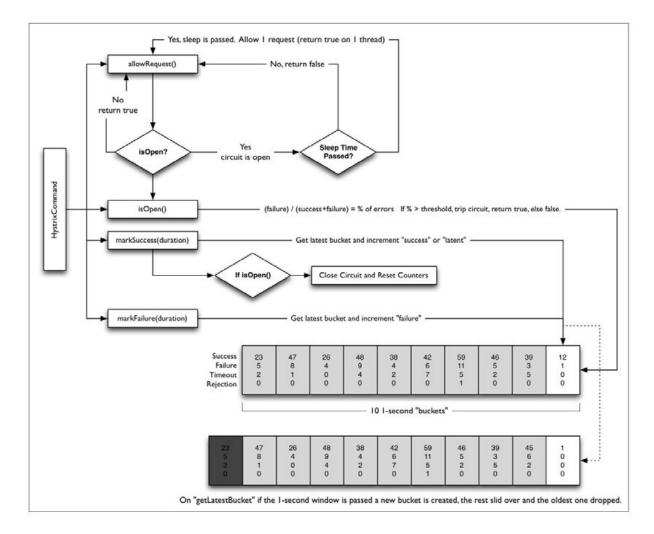
| ●toObservable□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
|---|
| |
| ●observe□□□□ toObservable□□□□□□ Observable □□□□□□ |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| ● queue 🛮 🗎 🖟 toObservable 🖺 🗎 🗎 🗎 🗎 Observable 🖺 🖺 |
| toBlocking |
| |
| ●execute queue Future get |
| |
| |
| |
| HystrixCommand HystrixObservableCommand |
| |
| |
| public interface HystrixCircuitBreaker { |
| public static class Factory {} |
| static class HystrixCircuitBreakerImpl implements |
| HystrixCircuitBreaker {} |
| static class NoOpCircuitBreaker implements |
| HystrixCircuitBreaker {} |
| public boolean allowRequest□□; |
| public boolean isOpen□□; |
| void markSuccess□□; |
| } |
| |

| ●allowRequest□□□□□Hystrix□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
|---|
| ●isOpen |
| ●markSuccess□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| ●□□□Factory□□□□□□Hystrix□□□HystrixCircuitBreaker□□□□ |
| ☐ ConcurrentHashMap <string,hystrixcircuitbreaker></string,hystrixcircuitbreaker> |
| circuitBreakersByCommand [] [] String [] [] key [] [] |
| HystrixCommandKey Hystrix key |
| Hystrix 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 Hystrix Circuit Breaker 🗆 🗆 |
| ●□□□NoOpCircuitBreaker□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| HystrixCircuitBreakerImpl |
| HystrixCircuitBreaker |
| ■ HystrixCommandProperties properties 🛮 🗎 🗎 🗎 |
| HystrixCommand |
| ■HystrixCommandMetrics metrics□□□□ HystrixCommand |
| |
| ■AtomicBoolean circuitOpen□□□□□□□□□□□□□□□false□ |
| ■AtomicLong circuitOpenedOrLastTestedTime □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| HystrixCircuitBreakerImpl\\ HystrixCircuitBreaker\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ |
| |
| ●isOpen/ |
| ■00000000true00000true0000000000000000000 |
| metrics |
| |

```
□□□□□□circuitBreakerRequestVolumeThreshold□□□□□20□
   public boolean isOpen□□{
       if∏circuitOpen.get∏∏∏{
        return true:
       }
       HealthCounts health=metrics.getHealthCounts□□;
       if | health.get Total Requests | | | <
    properties.circuitBreakerRequestVolumeThreshold
 \square \square.get\square \square \square
        return false:
       }
       if | health.getErrorPercentage | | | <
    properties.circuitBreakerErrorThresholdPercentage
 \square \square.get\square \square \square{return false;
       } else {
        if∏circuitOpen.compareAndSet∏false,true∏∏{
          circuitOpenedOrLastTestedTime.set
       □System.currentTimeMillis□□□;
          return true:
        } else {
          return true;
```

```
}
    }
 properties
isOpen[][]| allowSingleTest[][][][][][][][][]! isOpen[][][][][]
_____allowSingleTest
@Override
  public boolean allowRequest∏∏{
   if properties.circuitBreakerForceOpen ___.get ____{{}}
    return false:
    }
   if [ properties.circuitBreakerForceClosed ] [ ].get [ ] [ ] { }
    isOpen∏∏;
    return true;
    }
   return!isOpen□□|| allowSingleTest□□;
  }
 -
circuitBreakerSleepWindowInMilliseconds
```

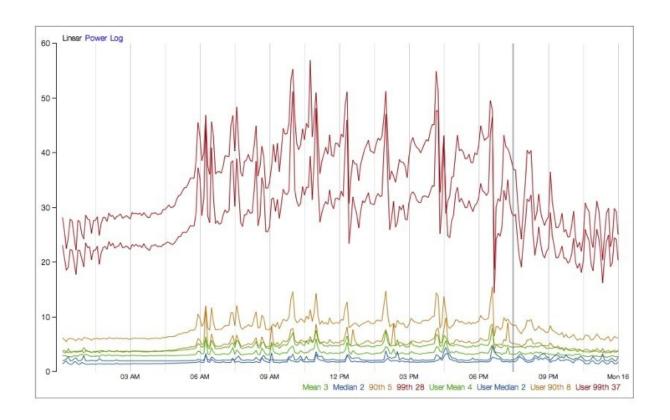
```
public boolean allowSingleTest□□{
       long
    timeCircuitOpenedOrWasLastTested=circuitOpenedOrL
    astTestedTime.get∏∏;
       if□circuitOpen.get□□&& System.currentTimeMillis□□
    >
       timeCircuitOpenedOrWasLastTested+
   properties.circuitBreakerSleepWindowInMilliseconds
\square \square .get \square \square \square \{
        if
     □circuitOpenedOrLastTestedTime.compareAndSet□
          time Circuit Opened Or Was Last Tested, System. cu\\
       rrentTimeMillis∏∏∏∏{
           return true:
        }
       return false:
    }
   public void markSuccess□□{
       if∏circuitOpen.get∏∏∏{
        if∏circuitOpen.compareAndSet∏true,false∏∏{
           metrics.resetStream∏;
```







| ullet |
|-------------------------------|
| |
| ullet |
| |
| ullet |
| |
| ullet |
| Spring |
| Cloud Config Spring Cloud Bus |
| ullet |
| |
| ullet |
| |
| |
| |
| |
| |
| |
| |
| Netflix |
| Onetflix Hystrix |
| |
| |



| 比较情况 | 未使用线程池隔离 | 使用了线程池隔离 | 耗时差距 |
|--------|----------|----------|------|
| 中位数 | 2ms | 2ms | 2ms |
| 90 百分位 | 5ms | 8ms | 3ms |
| 99 百分位 | 28ms | 37ms | 9ms |

| ●□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
|--|
| SEMAPHORE, Hystrix |
| ●□□□□□ □Hystrix□□□□□□□□□□□□□□□□□□□□ |
| |
| 00000000000000000000000000000000000000 |
| <pre>ps</pre> |
| |
| |
| |
| |
| |
| " " |
| |
| |
| |
| |
| |
| Hystrix DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| |
| public class UserCommand extends |
| HystrixCommand <user> {</user> |
| private RestTemplate restTemplate; |
| private Long id; |
| public UserCommand [] Setter setter,RestTemplate |
| restTemplate,Long id∏{ |

```
super[]setter[];
        this.restTemplate=restTemplate;
        this.id=id;
      }
      @Override
      protected User run∏∏{
        return restTemplate.getForObject [] "http://USER-
     SERVICE/users/{1}",User.class,id[];
      }
    }
  חחחחחחח UserCommand
                               UserCommand
     П
       П
          П
            П
               П
                   User
                        u=new
□restTemplate,1L□.execute□□;□
  get
  public class UserService {
      @Autowired
      private RestTemplate restTemplate;
      @HystrixCommand
      public User getUserById∏Long id∏{
        return restTemplate.getForObject [] "http://USER-
     SERVICE/users/{1}",User.class,id∏;
      }
    }
```

```
@HystrixCommand
                     public Future<User> getUserByIdAsync∏final String
           id∏{
                          return new AsyncResult<User>∏{
                                @Override
                                public User invoke□□{
                                                                                             restTemplate.getForObject
                                           return
                                □"http://USER-SERVICE/users/{1}",
                User.class,id∏;
                                }
                           };
           ∏∏Observable
                Observable<String>
                                                                                                                             UserCommand
                                                                                      ho=new
     □restTemplate,1L□.observe□□;
                Observable < String >
                                                                                                                             UserCommand
                                                                                       co=new
     □restTemplate,1L□.toObservable□□;
           On the control of the
Observable Cold Observable
toObservable
```

| HystrixCommand |
|----------------|
| OBOBServable |
| |
| Observable [|

```
public class UserObservableCommand extends HystrixObservableCommand<User> {
       private RestTemplate restTemplate;
       private Long id;
       public UserObservableCommand(Setter setter, RestTemplate restTemplate, Long id) {
          super (setter);
          this.restTemplate = restTemplate;
          this.id = id;
       @Override
       protected Observable<User> construct() {
          return Observable.create(new Observable.OnSubscribe<User>() {
             public void call(Subscriber<? super User> observer) {
                    if (!observer.isUnsubscribed()) {
                       User user = restTemplate.getForObject("http://USER-SERVICE/
users/{1}", User.class, id);
                       observer.onNext(user);
                       observer.onCompleted();
                 } catch (Exception e) {
                    observer.onError(e);
           } );
```

```
public Observable<User> getUserById ☐ final String
   id∏{
                    Observable.create
                                         П
        return
                                                new
     Observable.OnSubscribe<User>\square[
          @Override
          public void call □ Subscriber<? super User>
      observer∏{
             try {
               if∏! observer.isUnsubscribed∏∏{
                        user=restTemplate.getForObject
               □"http://HELLO-SERVICE/
     users/{1}",User.class,id∏;
                  observer.onNext∏user∏;
                  observer.onCompleted∏;
               }
             } catch∏Exception e∏{
               observer.onError∏e∏;
             }
          }
        }□;
      }
   observableExecutionMode ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ Observe ☐ ☐ ☐ ☐
toObservable
                                   @HystrixCommand
□ observableExecutionMode=ObservableExecutionMode.EAG
ER[]:EAGER[][][][][][][][][][]observe[][][][][]
```

@HystrixCommand □observableExecutionMode=ObservableExecutionMode.LAZY |||||||||toObservable|||||||| ___ run_____getFallback___ public class UserCommand extends HystrixCommand<User> { private RestTemplate restTemplate; private Long id; public UserCommand ☐ Setter setter,RestTemplate restTemplate,Long id□{ super[]setter[]; this.restTemplate=restTemplate; this.id=id; @Override protected User run□□{ restTemplate.getForObject ☐ "http://USER-SERVICE/users/{1}",User.class,id□; } @Override

protected User getFallback□□{

```
return new User∏;
   }
  }
 ☐ HystrixObservableCommand ☐☐ Hystrix ☐☐☐☐☐☐☐☐☐☐☐☐
fallbackMethod
  public class UserService {
    @Autowired
    private RestTemplate restTemplate;
    @HystrixCommand
  □fallbackMethod="defaultUser"□
    public User getUserById∏Long id∏{
     return restTemplate.getForObject □ "http://USER-
   SERVICE/users/{1}",User.class,id∏;
    }
    public User defaultUser□□{
     return new User∏;
    }
   }
 public∏∏∏
 ____defaultUser____getUserById______
```

```
public class UserService {
     @Autowired
     private RestTemplate restTemplate;
     @HystrixCommand
   ∏fallbackMethod="defaultUser"

☐
     public User getUserById

☐Long id

☐{
      return restTemplate.getForObject □ "http://USER-
    SERVICE/users/{1}",User.class,id□;
     }
     @HystrixCommand
   □fallbackMethod="defaultUserSec"□
     public User defaultUser∏∏{
      return new User∏"First Fallback"∏;
     }
     public User defaultUserSec□□{
      return new User∏"Second Fallback"∏;
     }
   }
```

| ●000000000000000000000000000000000000 |
|--|
| |
| |
| Hystrix |
| |
| |
| |
| |
| HystrixCommand run |
| HystrixBadRequestException |
| |
| Hystrix |
| @HystrixCommand[][]ignoreExceptions[][][][] |
| @HystrixCommand [] ignoreExceptions= |
| $\{BadRequestException.class\}$ |
| public User getUserById□Long id□{ |
| return restTemplate.getForObject [] "http://USER- |
| SERVICE/users/{1}",User.class,id□; |
| } |
| |
| $\verb $ |
| fallback |
| |
| <pre> Hystrix </pre> |
| |

| getFallback |
|---|
| Throwable getExecutionException |
| |
| |
| fallback |
| |
| @HystrixCommand[]fallbackMethod="fallback1"[] |
| User getUserById□String id□{ |
| throw new RuntimeException [] "getUserById |
| command failed"[]; |
| } |
| User fallback1□String id,Throwable e□{ |
| assert "getUserById command failed".equals |
| □e.getMessage□□□; |
| } |
| |
| |
| HystrixSetter |
| |
| public UserCommand□□{ |
| super Setter.withGroupKey |
| HystrixCommandGroupKey.Factory.asKey |
| □"GroupName"□□ |
| .andCommandKey |
| ☐ HystrixCommandKey.Factory.asKey |
| □"CommandName"□□;□; |

| } |
|---|
| Setter |
| withGroupKey [][[][[][][][][] andCommandKey [][[][][][][] |
| <pre>Description:</pre> <pre>Description:</pre> <pre>Description:</pre> <pre>Setter</pre> <pre>Setter</pre> <pre>Description:</pre> <pre>Setter</pre> <pre>Description:</pre> <pre>Setter</pre> <pre>Description:</pre> <pre>Setter</pre> <pre>Setter</pre> <pre>Description:</pre> <pre>Setter</pre> <pre>Description:</pre> <pre>Setter</pre> <pre>Setter</pre> <pre>Description:</pre> <pre>Setter</pre> <pre>Setter</pre> <pre>Setter</pre> <pre>Setter</pre> <pre>Setter</pre> <pre>Setter</pre> <pre>Setter</pre> |
| GroupKey |
| 0000000Hystrix000000000000000000000000000000000000 |
| 00000000000000000000000000000000000000 |
| |
| |
| HystrixHystrix _ |
| HystrixThreadPoolKey |
| |
| public UserCommand□□{ |
| super Setter.withGroupKey |
| HystrixCommandGroupKey.Factory.asKey |
| □"CommandGroupKey"□□ |
| .andCommandKey |
| HystrixCommandKey.Factory.asKey |
| □"CommandKey"□□ |
| .andThreadPoolKey |
| HystrixThreadPoolKey.Factory.asKey |
| □"ThreadPoolKey"□□□; |
| } |
| |
| |
| |
| |

| HystrixCommand |
|---|
| <pre>Description:</pre> Description: Description: Description: Description: Description: Description: Description: Description: Description: Description: Description: Description: Description: Description: Des |
| @HystrixCommand $\ \ \ \ \ \ \ \ \ \ \ \ \ $ |
| threadPoolKey |
| |
| @HystrixCommand |
| <pre>[]commandKey="getUserById",groupKey="UserGroup",thre</pre> |
| adPoolKey="getUserByIdThread"[] |
| public User getUserById Long id { |
| return restTemplate.getForObject [] "http://USER- |
| SERVICE/users/{1}",User.class,id□; |
| } |
| |
| |
| |
| |
| |
| |
| |
| 000000000000000Hystrix000000000000000000000000000000000000 |
| |
| |
| Hystrix |
| HystrixObservableCommand [][][][] getCacheKey[][][][][][] |
| |

```
HystrixCommand<User> {
     private RestTemplate restTemplate;
     private Long id;
     public
                               RestTemplate
             UserCommand
                          П
  restTemplate,Long id∏{
      super
                          Setter.withGroupKey
             HystrixCommandGroupKey.Factory.asKey
   □"UserGroup"□□□;
      this.restTemplate=restTemplate;
      this.id=id:
     }
     @Override
     protected User run∏∏{
      return restTemplate.getForObject ☐ "http://USER-
   SERVICE/users/{1}",User.class,id□;
     }
     @Override
     protected String getCacheKey□□{
      return String.valueOf∏id∏;
     }
  ______etCacheKey_____key____
cacheKey_____
```

UserCommand

extends

public

class

```
ullet
   ullet \Pi
  public
           class
                 UserGetCommand
                               extends
  HystrixCommand<User> {
     private
                        HystrixCommandKey
            static
                  final
   GETTER KEY=HystrixCommandKey.Factory.asKey
   □"CommandKey"□;
      private RestTemplate restTemplate;
     private Long id;
     public
                            RestTemplate
            UserGetCommand
                         П
   restTemplate,Long id<sub>□</sub>{
       super
                 П
                        Setter.withGroupKey
            HystrixCommandGroupKey.Factory.asKey
    П
    □"GetSetGet"□□
        .andCommandKey□GETTER KEY□□;
       this.restTemplate=restTemplate;
       this.id=id;
      }
     @Override
     protected User run□□{
```

```
return restTemplate.getForObject [] "http://USER-
      SERVICE/users/{1}",User.class,id□;
        }
        @Override
        protected String getCacheKey□□{
          //NNidNNN
          return String.valueOf□id□;
        }
        public static void flushCache∏Long id∏{
          //_____id___
          HystrixRequestCache.getInstance∏GETTER KEY,
            HystrixConcurrencyStrategyDefault.getInstanc
        e \square \square .clear \square String.valueOf \square id \square :
      public
                 class
                           UserPostCommand
                                                    extends
  HystrixCommand<User> {
        private RestTemplate restTemplate;
        private User user;
        public
                  UserPostCommand
                                              RestTemplate
    restTemplate,User user [[{
                                       Setter.withGroupKey
    super
                   HystrixCommandGroupKey.Factory.asKey
□"GetSetGet"□□□;
          this.restTemplate=restTemplate;
          this.user=user;
        }
```

```
@Override
     protected User run□□{
       User r=restTemplate.postForObject□"http://USER-
    SERVICE/users",user,User.class□;
       UserGetCommand.flushCache ☐ user.getId ☐ ☐ ☐ ☐ ;
       return r;
     }
  □□□□□□□□□□□□□□UserGetCommand □□□□ id □□ User □□□□
UserPostCommand \square \square \square User \square \square \square \square \square UserGetCommand \square \square \square \square \square
00Hystrix000000Key00000000000User000000000
_____U___U___UserGetCommand _______
flushCache [] [] [] [] HystrixRequestCache.getInstance
☐ GETTER KEY, HystrixConcurrencyStrategyDefault.getInstanc
ennanga Hystrix nggang GETTER KEYnnangangang
```

```
\square\square\squareAbstractCommand\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\squaregetCacheKey\square\square\square\square
abstract class AbstractCommand<R> implements
   HystrixInvokableInfo<R>,
    HystrixObservable<R> {
                     final
                              HystrixRequestCache
       protected
    requestCache;
       protected String getCacheKey∏∏{
         return null;
       }
       protected boolean isRequestCachingEnabled□□{
         return properties.requestCacheEnabled □ □.get □ □
      && getCacheKey∏∏!=null;
       }
       public Observable<R> toObservable∏∏{
         final
                                        boolean
      requestCacheEnabled=isRequestCachingEnabled;
```

```
final String cacheKey=getCacheKey□□;
        final AbstractCommand<R> cmd=this;
         if∏requestCacheEnabled∏{
          HystrixCommandResponseFromCache<R>
       fromCache=
         HystrixCommandResponseFromCache<R>
                                                    П
requestCache.get[]cacheKey[];
          if∏fromCache !=null∏{
            isResponseFromCache=true;
            return handleRequestCacheHitAndEmitValues
        □fromCache, cmd□;
          }
         }
         Observable<R> hystrixObservable=
   Observable.defer 

applyHystrixSemantics
                                                 .map
| wrapWithAllOnNextHooks|;
         Observable < R > after Cache:
        if request Cache Enabled & cache Key!=null {
          HystrixCachedObservable<R> toCache=
   HystrixCachedObservable.from
□hystrixObservable,this□;
          HystrixCommandResponseFromCache<R>
       fromCache=
         HystrixCommandResponseFromCache<R>
                                                     П
requestCache.putIfAbsent\( \text{cacheKey,toCache} \);
```

```
if□fromCache !=null□{
      toCache.unsubscribe∏;
      isResponseFromCache=true;
      return handleRequestCacheHitAndEmitValues
    []fromCache,_cmd[];
     } else {
      afterCache=toCache.toObservable∏;
     }
    } else {
     afterCache=hystrixObservable;
    }
    }
 ____getCacheKey _____null ___null ___ Key _____
getCacheKey [ ] Key [ ] HystrixRequestCache [ ] get
□String cacheKey□□□□□□□HystrixCachedObservable□□□
 hystrixObservable
hystrixObservable
```

| nullKeytoCache |
|---|
| |
| $handle Request Cache Hit And Emit Values \verb $ |
| fromCache null |
| Observable DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| Hystrix000000000000000000000000000000000000 |
| JSR 107Hystrix |
| JSR 10700000000000000000000000000000000000 |

| 注解 | 描述 | 属性 |
|--------------|---|----------------|
| @CacheResult | 该注解用来标记请求命令返回的结果应该被缓存,它必须与 | cacheKeyMethod |
| | @HystrixCommand 注解结合使用 | |
| @CacheRemove | 该注解用来让请求命令的缓存失效,失效的缓存根据定义的 | commandKey, |
| | Key 决定 | cacheKeyMethod |
| @CacheKey | 该注解用来在请求命令的参数上标记,使其作为缓存的 Key | value |
| | 值,如果没有标注则会使用所有参数。如果同时还使用了 | |
| | @CacheResult 和@CacheRemove 注解的 cacheKeyMethod | |
| | 方法指定缓存 Key 的生成,那么该注解将不会起作用 | |

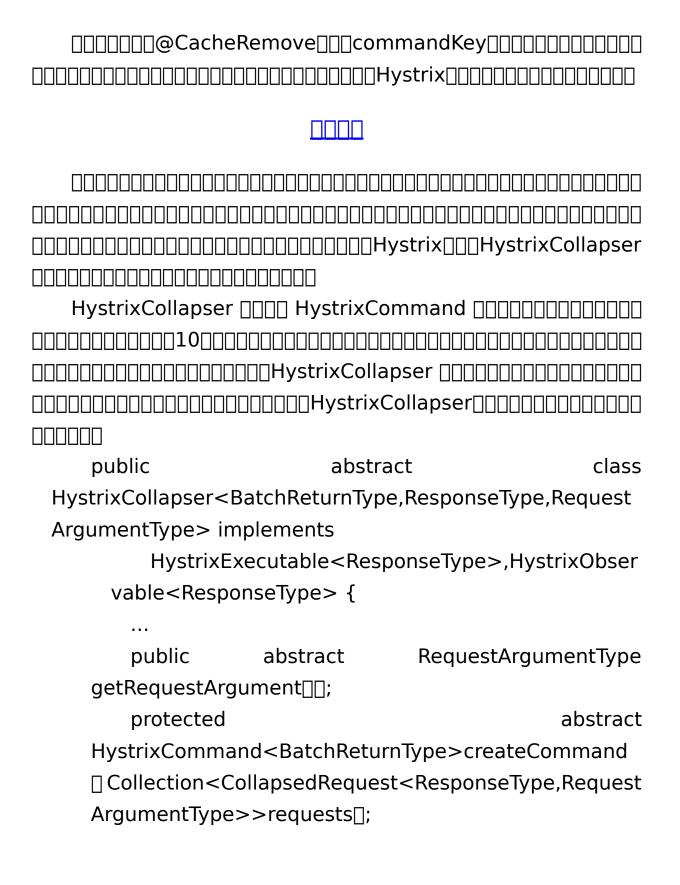
- - @CacheResult
 - @HystrixCommand
 public User getUserById
 Long id
 {

```
restTemplate.getForObject [] "http://USER-
    SERVICE/users/{1}",User.class,id□;
    }
   \square\square\square\square\square@CacheResult\square@CacheRemove\square\square\squarecacheKeyMethod\square
nnnnnnnnnnnnn@CacheKeynnnnnnnnnnnKeynnn
   Key = 0 = 0 = 0 CacheResult 0 = 0 CacheKeyMethod 0 = 0 = 0
□□□□□@HystrixCommand□□□□fallbackMethod□□□□
    @CacheResult
 □cacheKeyMethod="getUserByIdCacheKey"□
    @HystrixCommand
    public User getUserById

Long id

{
       return restTemplate.getForObject ☐ "http://USER-
    SERVICE/users/{1}",User.class,id∏;
    }
    private Long getUserByIdCacheKey∏Long id∏{
       return id;
    }
   @CacheKey[][][][][][][][][][][][]cacheKeyMethod[][][][][][][][]
@CacheResult
      @HystrixCommand
      public User getUserById∏@CacheKey∏"id"∏Long id∏{
             restTemplate.getForObject □ "http://USER-
       return
    SERVICE/users/{1}",
```

```
User.class,id∏;
  @CacheResult
   @HystrixCommand
   public User getUserById @CacheKey "id" User user {
          restTemplate.getForObject [] "http://USER-
   SERVICE/users/{1}",User.class,user.getId∏∏;
   }
  Hystrix
@CacheResult
   @HystrixCommand
   public User getUserById□@CacheKey□"id"□Long id□{
          restTemplate.getForObject ☐ "http://USER-
   SERVICE/users/{1}",User.class,id□;
   }
   @CacheRemove[]commandKey="getUserById"[]
   @HystrixCommand
   public void update[@CacheKey["id"[User user[]{
     return restTemplate.postForObject ☐ "http://USER-
   SERVICE/users",user,User.class□;
   }
```



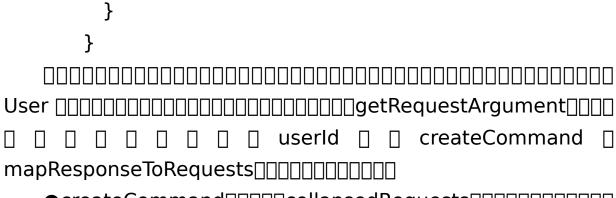
| protected abstract void mapResponseToRequests |
|--|
| ☐ BatchReturnType |
| batchResponse,Collection <collapsedrequest<respons< td=""></collapsedrequest<respons<> |
| eType,RequestArgumentType>> requests[]; |
| *** |
| } |
| |
| ●BatchReturnType□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| ●ResponseType□□□□□□□□□□ |
| ■RequestArgumentType□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| ■RequestArgumentType getRequestArgument□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| HystrixCommand <batchreturntype>createCommand</batchreturntype> |
| Collection < CollapsedRequest < ResponseType, RequestArgu |
| mentType>> requests[][][][][][][][][][][][][][][][][][][|
| ■ mapResponseToRequests □ BatchReturnType |
| batchResponse,Collection <collapsedrequest<responsetype< td=""></collapsedrequest<responsetype<> |
| RequestArgumentType>> requests |
| |
| |
| |
| ●/users/{id}idUserGET |
| ●/users? ids={ids}idsUserGETids |
| |
| RestTemplate |
| |

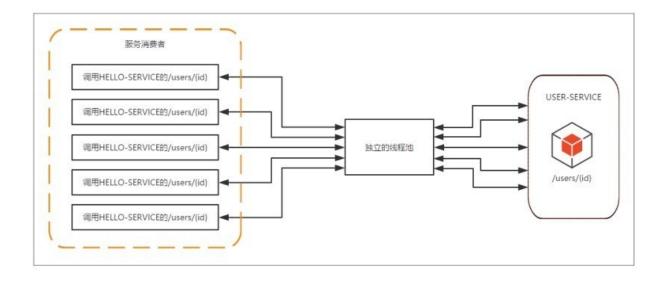
```
public class UserServiceImpl implements UserService {
       @Autowired
       private RestTemplate restTemplate;
       @Override
       public User find \text{\pilot} Long id \text{\pilot} {
        return restTemplate.getForObject [] "http://USER-
     SERVICE/users/{1}",User.class,id∏;
       }
       @Override
       public List<User> findAll□List<Long> ids□{
        return restTemplate.getForObject □ "http://USER-
                      ids={1}",List.class,StringUtils.join
     SERVICE/users?
     □ids,","□□;
 ullet
                      UserBatchCommand
   public
             class
                                              extends
HystrixCommand<List<User>> {
     UserService userService:
     List<Long> userIds:
                                     П
                                          UserService
               UserBatchCommand
     public
 userService,List<Long> userIds[]{
                    Setter.withGroupKey
                                                asKey
       super
               П
                                           П
   □"userServiceCommand"□□□;
       this.userIds=userIds:
```

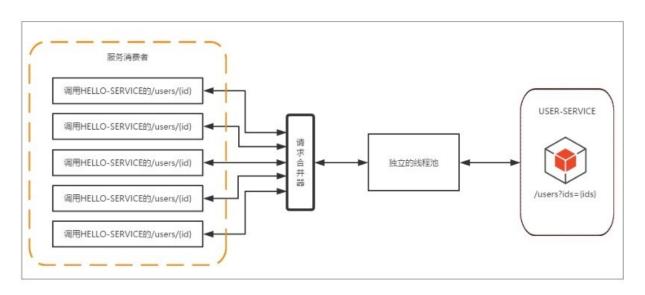
@Service

```
this.userService=userService:
       }
       @Override
       protected List<User> run throws Exception {
         return userService.findAll[]userIds[];
       }
      }
   □□□ userService.findAll □□□□□/users? ids={ids}□□□□□User□□□□
● □ □ □ □ □ □ □ HystrixCollapser □ □ □ □ □ □ □
       public
                         UserCollapseCommand
                                                  extends
                class
   HystrixCollapser<List<User>,User,Long> {
         private UserService userService;
         private Long userId;
                                           □ UserService
                  UserCollapseCommand
         public
     userService,Long userId∏{
                                   Setter.withCollapserKey
                         super
       ☐ HystrixCollapserKey. Factory. as Key
     □"userCollapseCommand"□□.andCollapserPropertiesDe
  faults□
     HystrixCollapserProperties.Setter
  \square\square.withTimerDelayInMilliseconds\square100\square\square\square;
           this.userService=userService:
           this.userId=userId;
         }
         @Override
```

```
public Long getRequestArgument□□{
         return userId;
       }
       @Override
       protected HystrixCommand<List<User>>
   createCommand
              Collection < Collapsed Request < User, Long >>
collapsedRequests[]{
         List<Long>
                                              ArrayList<>
                           userlds=new
     □collapsedRequests.size□□□;
   userIds.addAll ☐ collapsedRequests.stream ☐ ☐ .map
□CollapsedRequest::getArgument□.collec
   t \cap Collectors.toList \cap \cap \cap \cap;
                                      UserBatchCommand
         return
                         new
     ∏userService,userIds∏;
        }
       @Override
       protected
                                 mapResponseToRequests
                       void
   □List<User> batchResponse,
   Collection < Collapsed Request < User, Long >>
collapsedRequests[]{
         int count=0;
                           CollapsedRequest<User,Long>
         for
     collapsedRequest : collapsedRequests□{
           User user=batchResponse.get\squarecount++\square;
           collapsedRequest.setResponse∏user∏;
          }
```







```
@HystrixProperty
                              □name="timerDelayInMilliseconds",value="100"□
                               }∏
                               public User find Long id { {
                                      return null;
                               }
                               @HystrixCommand
                               public List<User> findAll_List<Long> ids[]{
                                                                    restTemplate.getForObject ☐ "http://USER-
                                      return
                                                                                                        ids={1}",List.class,StringUtils.join
                       SERVICE/users?
                       ∏ids,","∏∏;
                               }
                       }
               _____@HystrixCommand _______
\label{localization} $$ \Box \Box \Box \Delta = \frac{1}{2} \int_{\mathbb{R}^{n}} |\Delta u = \frac{1}{2} \int_{\mathbb{R}^{n}} |
{ids}_____collapserProperties______
@HystrixProperty
100____@HystrixCollapser________/users/{id}___
```

| ullet |
|---|
| |
| ullet |
| |
| |
| |
| |
| |
| |
| |
| HystrixHystrixHystrix |
| HystrixCommand |
| ●□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| public HystrixCommandInstance□int id□{ |
| super Setter.withGroupKey |
| ☐ HystrixCommandGroupKey.Factory.asKey |
| □"ExampleGroup"□□ |
| . and Command Properties Defaults |
| □HystrixCommandProperties.Setter |
| $. with {\sf Execution Timeout In Millise conds}$ |
| □500□□□; |
| this.id=id; |
| } |
| ● |
| commandProperties[][][][][] |

```
@HystrixCommand[commandKey="helloKey",
                        commandProperties={
                             @HystrixProperty
                   □ name="execution.isolation.thread.timeoutInMillisec
                   onds",value="5000"∏
              public User getUserById

☐Long id

☐{
                                           restTemplate.getForObject ☐ "http://USER-
                        return
              SERVICE/users/{1}",User.class,id□;
               }
          ullet
Config Spring Cloud Bus Config Spring Cloud Bus Config Con
ullet
Spring Cloud Config Spring Cloud Bus
```

Command □□

| CommandHystrixCommand |
|--|
| |
| execution□□ |
| execution |
| execution.isolation.strategy |
| HystrixCommand.run |
| |
| |
| ■SEMAPHORE |
| |
| |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|--|
| 全局默认值 | THREAD |
| 全局配置属性 | hystrix.command.default.execution.isolation.strategy |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withExecutionIsolationStrategy (ExecutionIsolationStrategy.THREAD) 设置,也可通过@HystrixProperty (name="execution.isolation.strategy", value= "THREAD")注解设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.execution.isolation.strategy |

| 属性级别 | 默认值、配置方式、配置属性 |
|-------|---------------|
| 全局默认值 | 1000 毫秒 |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|--|
| 全局配置属性 | hystrix.command.default.execution.isolation.thread.timeoutIn- |
| | Milliseconds |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withExecutionTimeoutIn- |
| | Milliseconds(int value)设置,也可通过@HystrixProperty(name= |
| | "execution.isolation.thread.timeoutInMilliseconds",value="2000") |
| | 注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.execution.isolation.thread. |
| | timeoutInMilliseconds |

| execution.timeout.enabled | | | | | | | |
|---|------|------|-----|-----|-------|------|--|
| HystrixCommand.run | |]t | rue | |]fal: | se[] | |
| execution. is olation. thread. time out Irrupation and the state of | Milم | lise | con | ds[| | | |
| | | | | | | | |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局默认值 | true |
| 全局配置属性 | hystrix.command.default.execution.timeout.enabled |
| 实例默认值 | 通 过 HystrixCommandProperties.Setter().withExecutionTimeout-
Enabled(boolean value) 设置,也可通过@HystrixProperty(name= |
| | "execution.timeout.enabled", value="false")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.execution.timeout.enabled |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|--|
| 全局默认值 | true |
| 全局配置属性 | hystrix.command.default.execution.isolation.thread. |
| | interruptOnTimeout |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withExecutionIsolation- |
| | ThreadInterruptOnTimeout(boolean value)设置,也可通过@Hystrix- |
| | Property(name="execution.isolation.thread.interruptOnTimeout", |
| | value="false")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.execution.isolation.thread. |
| | interruptOnTimeout |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局默认值 | true |
| 全局配置属性 | hystrix.command.default.execution.isolation.thread. |
| | interruptOnCancel |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withExecutionIsolation- |
| | ThreadInterruptOnCancel(boolean value)设置,也可通过@Hystrix- |
| | Property(name="execution.isolation.thread.interruptOnCancel", |
| | value="false")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.execution.isolation.thread. |
| | interruptOnCancel |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|--|
| 全局默认值 | 10 |
| 全局配置属性 | hystrix.command.default.execution.isolation.semaphore. |
| | maxConcurrentRequests |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withExecutionIsolation- |
| | SemaphoreMaxConcurrentRequests(int value)设置,也可通过@Hystrix- |
| | Property(name="execution.isolation.semaphore.maxConcurrentRe |
| | quests", value="2")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.execution.isolation.semaphore. |
| | maxConcurrentRequests |

fallback□□

| 000000000000000000000000000000000000000 | []fallback |
|---|------------|
| | |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|--|
| 全局默认值 | 10 |
| 全局配置属性 | hystrix.command.default.fallback.isolation.semaphore. |
| | maxConcurrentRequests |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withFallbackIsolation- |
| | SemaphoreMaxConcurrentRequests(int value)设置,也可通过@HystrixProperty |
| | (name="fallback.isolation.semaphore.maxConcurrentRequests", |
| | value="20")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.fallback.isolation.semaphore. |
| | maxConcurrentRequests |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|--|
| 全局默认值 | true |
| 全局配置属性 | hystrix.command.default.fallback.enabled |
| 实例默认值 | 通 过 HystrixCommandProperties.Setter().withFallbackEnabled (boolean value)设置,也可通过@HystrixProperty(name= "fallback.enabled", value="false")注解来设置 |
| 实例配置属性 | 配置属性: hystrix.command.HystrixCommandKey.fallback.enabled |

circuitBreaker \square

| ●circuitBreaker.enable | d | | | | |
|------------------------|---|--|--|--|--|
| | | | | | |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局默认值 | true |
| 全局配置属性 | hystrix.command.default.circuitBreaker.enabled |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withCircuitBreakerEnabled (boolean value)设置,也可通过@HystrixProperty(name="circuitBreaker.enabled",value="false")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.circuitBreaker.enabled |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局默认值 | 20 |
| 全局配置属性 | hystrix.command.default.circuitBreaker.requestVolumeThreshold |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withCircuitBreakerRequest- |
| | VolumeThreshold(int value)设置,也可通过@HystrixProperty(name= |
| | "circuitBreaker.requestVolumeThreshold", value="30")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.circuitBreaker. |
| | requestVolumeThreshold |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|--|
| 全局默认值 | 5000 |
| 全局配置属性 | hystrix.command.default.circuitBreaker.sleepWindowInMilliseconds |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withCircuitBreakerSleep- |
| | WindowInMilliseconds(int value)设置,也可通过@HystrixProperty |
| | (name="circuitBreaker.sleepWindowInMilliseconds",value="3000") |
| | 注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.circuitBreaker. |
| | sleepWindowInMilliseconds |

| •circuitBreaker.errorThresholdPercentage | | | |
|--|--|--|--|
| 000000000000000000000000000000000000000 | | | |

| circuitBreaker.requestVolumeThreshold | 000000000000000000000000000000000000000 |
|---------------------------------------|---|
| | |

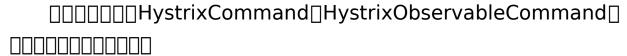
| 属性级别 | 默认值、配置方式、配置属性 |
|-------|---------------|
| 全局默认值 | 50 |

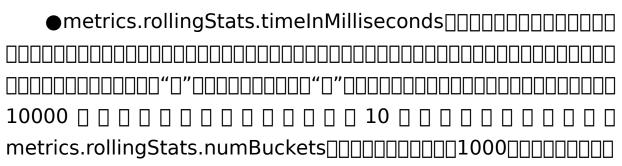
| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局配置属性 | hystrix.command.default.circuitBreaker.errorThresholdPercentage |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withCircuitBreakerError-ThresholdPercentage(int value)设置,也可通过@HystrixProperty(name="circuitBreaker.errorThresholdPercentage", value="40")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.circuitBreaker. errorThresholdPercentage |

| 属性级别 | 默认值、配置方式、配置属性 | | | | | |
|--------|--|--|--|--|--|--|
| 全局默认值 | false | | | | | |
| 全局配置属性 | hystrix.command.default.circuitBreaker.forceOpen | | | | | |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withCircuitBreakerForceOpen | | | | | |
| | (boolean value)设置,也可通过@HystrixProperty (name="circuitBreaker. | | | | | |
| | forceOpen", value="true")注解来设置 | | | | | |
| 实例配置属性 | hystrix.command.HystrixCommandKey.circuitBreaker. forceOpen | | | | | |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局默认值 | false |
| 全局配置属性 | hystrix.command.default.circuitBreaker.forceClosed |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withCircuitBreakerForce- |
| | Closed(boolean value) 设置,也可通过@HystrixProperty(name= |
| | "circuitBreaker.forceClosed", value="true")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.circuitBreaker.forceClosed |

metrics []





| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局默认值 | 10000 |
| 全局配置属性 | hystrix.command.default.metrics.rollingStats.timeInMilliseconds |
| 实例默认值 | 通 过 HystrixCommandProperties.Setter().withMetricsRolling-StatisticalWindowInMilliseconds(int value) 设置, 也可通过@HystrixProperty(name="metrics.rollingStats.timeInMilliseconds",value="20000")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.metrics.rollingStats.timeIn- |
| | Milliseconds |

| | □ H y | yst | rix | x 1 | 4 | .1 | 2[| | Br | ΉX | to | n. | S | R5 | 5 <u> </u> | | 1. | 5. | .3 | |
|--|--------------|-----|-----|------------|---|----|----|--|----|----|----|----|---|----|------------|--|----|----|----|--|
| | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |

| ●metri | cs.rollingStats | s.numBuck | cets[][][] | | | |
|------------|-----------------|-----------|------------|--|--|--|
| 000"0"0000 |] | | | | | |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|--|
| 全局默认值 | 10 |
| 全局配置属性 | hystrix.command.default.metrics.rollingStats.numBuckets |
| 实例默认值 | 通 过 HystrixCommandProperties.Setter().withMetricsRolling-StatisticalWindowBuckets(int value) 设置, 也可通过@Hystrix-Property(name="metrics.rollingStats.numBuckets",value="20")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.metrics.rollingStats. |
| | numBuckets |

| <pre> metrics.rollingStats.timeInMilliseconds </pre> |
|--|
| <pre>Description</pre> |
| metrics.rollingStats.timeInMilliseconds 🛮 🗎 🗎 🗎 Hystrix |
| 1.4.12 |
| |
| ●metrics.rollingPercentile.enabled ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ |
| 000000000000000false00000000000-10 |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|--|
| 全局默认值 | true |
| 全局配置属性 | hystrix.command.default.metrics.rollingPercentile.enabled |
| 实例默认值 | 通 过 HystrixCommandProperties.Setter().withMetricsRolling-PercentileEnabled(boolean value) 设置,也可通过@HystrixProperty (name="metrics.rollingPercentile.enabled", value="false")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.metrics.rollingPercentile. |
| | enabled |

• metrics.rollingPercentile.timeInMilliseconds

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局默认值 | 60000 |
| 全局配置属性 | hystrix.command.default.metrics.rollingPercentile. timeInMilliseconds |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withMetricsRolling-PercentileWindowInMilliseconds(int value) 设置, 也可通过@HystrixProperty(name="metrics.rollingPercentile.timeInMilliseconds", value="50000")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.metrics.rollingPercentile.timeInMilliseconds |

| |][] [] H } | /strix | < 1.4 | .12 | Brix | ton. | SR5[| 001. | 5.3 |
|---|-------------------|--------|-----------------|-----|------|------|------|-------------|-----|
| | | | | | | | | | |
| П | | | | | | | | | |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|--|
| 全局默认值 | 6 |
| 全局配置属性 | hystrix.command.default.metrics.rollingPercentile.numBuckets |
| 实例默认值 | 通 过 HystrixCommandProperties.Setter().withMetricsRolling-
PercentileWindowBuckets(int value)设置,也可通过@HystrixProperty
(name="metrics.rollingPercentile.numBuckets",value="5")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.metrics.rollingPercentile. |
| | numBuckets |

| BHystrix 1.4.12Brixton.SR51.5.3 |
|---------------------------------|
| |
| |

| ●metrics.rollingPercentile.bucketSize□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
|---|
| "a"aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa |
| |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|--|
| 全局默认值 | 100 |
| 全局配置属性 | hystrix.command.default.metrics.rollingPercentile.bucketSize |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withMetricsRollingPercentile- |
| | BucketSize(int value)设置,也可通过@HystrixProperty(name="metrics. |
| | rollingPercentile.bucketSize",value= "120")注解来设置 |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|--|
| 实例配置属性 | hystrix.command.HystrixCommandKey.metrics.rollingPercentile. |
| | bucketSize |

•metrics.healthSnapshot.intervalInMilliseconds

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局默认值 | 500 |
| 全局配置属性 | hystrix.command.default.metrics.healthSnapshot.intervalIn- |
| | Milliseconds |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withMetricsHealthSnapshot- |
| | IntervalInMilliseconds(int value)设置,也可通过@HystrixProperty- |
| | (name="metrics.healthSnapshot.intervalInMilliseconds", value= |
| | "600")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.metrics.healthSnapshot. |
| | intervalInMilliseconds |

requestContext[[[

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|--|
| 全局默认值 | true |
| 全局配置属性 | hystrix.command.default.requestCache.enabled |
| 实例默认值 | 通过 HystrixCommandProperties.Setter().withRequestCacheEnabled (boolean value)设置,也可通过@HystrixProperty(name="requestCache.enabled", value="false")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.requestCache.enabled |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局默认值 | true |
| 全局配置属性 | hystrix.command.default.requestLog.enabled |
| 实例默认值 | 通 过 HystrixCommandProperties.Setter().withRequestLogEnabled |
| | (boolean value)设置,也可通过@HystrixProperty(name="requestLog. |
| | enabled", value="false")注解来设置 |
| 实例配置属性 | hystrix.command.HystrixCommandKey.requestLog.enabled |

<u>collapser</u>∏

| @HystrixCollapser[] collapserProperties[] [] [] [] [] [] |
|--|
| @HystrixCollapser |
| <pre>[batchMethod="batch",collapserProperties={</pre> |
| @HystrixProperty |
| <pre>name="timerDelayInMilliseconds",value="20"</pre> |
| }□ |
| |
| ●maxRequestsInBatch□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局默认值 | Integer.MAX_VALUE |
| 全局配置属性 | hystrix.collapser.default.maxRequestsInBatch |
| 实例默认值 | 通过 HystrixCollapserProperties.Setter().withMaxRequestsInBatch (int value)设置,也可通过@HystrixProperty(name="maxRequestsInBatch", value="false")注解来设置 |
| 实例配置属性 | hystrix.collapser.HystrixCollapserKey.maxRequestsInBatch |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|--|
| 全局默认值 | 10 |
| 全局配置属性 | hystrix.collapser.default.timerDelayInMilliseconds |
| 实例默认值 | 通 过 HystrixCollapserProperties.Setter().withTimerDelayIn-Milliseconds(int value)设置,也可通过@HystrixProperty(name="timerDelayInMilliseconds",value="20")注解来设置 |
| 实例配置属性 | hystrix.collapser.HystrixCollapserKey.timerDelayInMilliseconds |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|--|
| 全局默认值 | true |
| 全局配置属性 | hystrix.collapser.default.requestCache.enabled |
| 实例默认值 | 通过 HystrixCollapserProperties.Setter().withRequestCacheEnabled |
| | (boolean value)设置,也可通过@HystrixProperty(name="requestCache. |
| | enabled", value="false")注解来设置 |
| 实例配置属性 | hystrix.collapser.HystrixCollapserKey.requestCache. enabled |

threadPool

| @HystrixCommand[][]threadPoolProperties[][][][][][][] |
|---|
| @HystrixCommand |
| <pre>[]fallbackMethod="helloFallback",commandKey="helloKey"</pre> |
| threadPoolProperties={ |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局默认值 | 10 |
| 全局配置属性 | hystrix.threadpool.default.coreSize |
| 实例默认值 | 通过 HystrixThreadPoolProperties.Setter().withCoreSize(int value) |
| | 设置,也可通过@HystrixProperty(name="coreSize", value="false")注 |
| | 解来设置 |
| 实例配置属性 | hystrix.threadpool.HystrixThreadPoolKey.coreSize |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局默认值 | 1: |
| 全局配置属性 | hystrix.threadpool.default.maxQueueSize |

属性级别默认值、配置方式、配置属性实例默认值通过 HystrixThreadPoolProperties.Setter().withMaxQueueSize (int value) 设置, 也可通过@HystrixProperty(name="maxQueueSize", value="10")注解来设置实例配置属性hystrix.threadpool.HystrixThreadPoolKey.maxQueueSize

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局默认值 | 5 |
| 全局配置属性 | hystrix.threadpool.default.queueSizeRejectionThreshold |
| 实例默认值 | 通过 HystrixThreadPoolProperties.Setter().withQueueSizeRejection-
Threshold(int value) 设置, 也可通过@HystrixProperty(name="queueSizeRejectionThreshold", value="10")注解来设置 |
| 实例配置属性 | hystrix.threadpool.HystrixThreadPoolKey.queueSizeRejection-
Threshold |

•metrics.rollingStats.timeInMilliseconds

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局默认值 | 10000 |
| 全局配置属性 | hystrix.threadpool.default.metrics.rollingStats.time- |
| | InMilliseconds |
| 实例默认值 | 通过 HystrixThreadPoolProperties.Setter().withMetricsRolling- |
| | StatisticalWindowInMilliseconds(int value)设置,也可通过@HystrixProperty |
| | (name="metrics.rollingStats.timeInMilliseconds", value="10")注 |
| | 解来设置 |

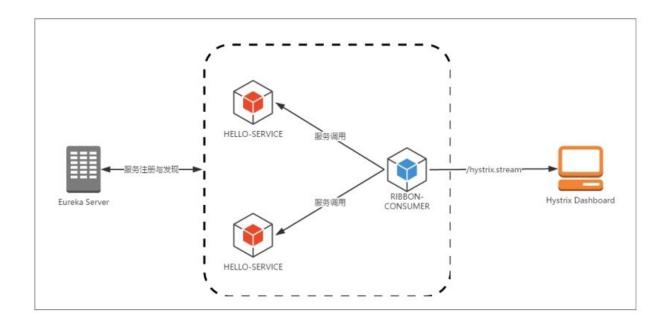
| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 实例配置属性 | hystrix.threadpool.HystrixThreadPoolKey.metrics.rollingStats. |
| | timeInMilliseconds |

| 属性级别 | 默认值、配置方式、配置属性 |
|--------|---|
| 全局默认值 | 10 |
| 全局配置属性 | hystrix.threadpool.default.metrics.rollingStats. numBuckets |
| 实例默认值 | 通 过 HystrixThreadPoolProperties.Setter().withMetricsRolling-
StatisticalWindowBuckets(int value) 设置, 也可通过@HystrixProperty
(name="metrics.rollingStats.numBuckets", value="10")注解来设置 |
| 实例配置属性 | hystrix.threadpool.HystrixThreadPoolKey.metrics.rollingStats. numBuckets |

| | 5 000000 |
|----------------------------------|-----------------|
| metrics.rolling-Stats.numBuckets | |

Hystrix

| UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUU |
|--|
| Hystrix |
| 0000"0"0"00000000000000000000000000000 |
| |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Spring Cloud [[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[|
| Hystrix Dashboard Dominic Dashboard |
| |
| |
| RIBBON-CONSUMER DO DO DO DO DO DO DO DE DE CONSUMER DO DO DO DO DO DO DO DE CONSUMER DO DO DO DO DO DO DO DO D |



□Spring Cloud□□□□□Hystrix Dashboard□□□□□□□□□□4□□

- ●□□□□□□Spring Boot□□□□□hystrix-dashboard□
- ●□□pom.xml□□□□□□□□□□

<parent>

<groupId>org.springframework.cloud

<artifactId>spring-cloud-starter-

parent</artifactId>

<version>Brixton.SR5</version>

<relativePath /> <!--lookup parent from repository-

->

</parent>

<dependencies>

<dependency>

<groupId>org.springframework.cloud

<artifactId>spring-cloud-starter-

hystrix</artifactId>

</dependency>

```
<dependency>
        <groupId>org.springframework.cloud
        <artifactId>spring-cloud-starter-hystrix-
     dashboard</artifactId>
       </dependency>
       <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-
     actuator</artifactId>
       </dependency>
    </dependencies>
  ■ □ □ □ □ □ □ □ @EnableHystrixDashboard □ □ □ Hystrix
Dashboard∏∏
  spring.application.name=hystrix-dashboard
    server.port=2001
  http://localhost:2001/hystrix
```

| Hystrix Dashboard http://hostname.port/turbine/turbine.stream |
|---|
| Cluster via Turbine (default cluster): http://turbine-hostname:port/turbine.stream Cluster via Turbine (custom cluster): http://turbine-hostname:port/turbine.stream?cluster= [clusterName] Single Hystrix App: http://hystrix-app:port/hystrix.stream Delay: 2000 ms Title: Example Hystrix App Monitor Stream |
| Hystrix Dashboard |
|]Hystrix Dashboard |
| • |
| nostname:port/turbine.stream[][][][][][][][][][][] |
| ● □ □ □ □ □ □ □ URLhttp://turbine- |
| nostname:port/turbine.stream? |
|]clusterName[][][][] |
| ●□□□□□□□□ □□ URLhttp://hystrix-app:port/hystrix.stream□ |
| |

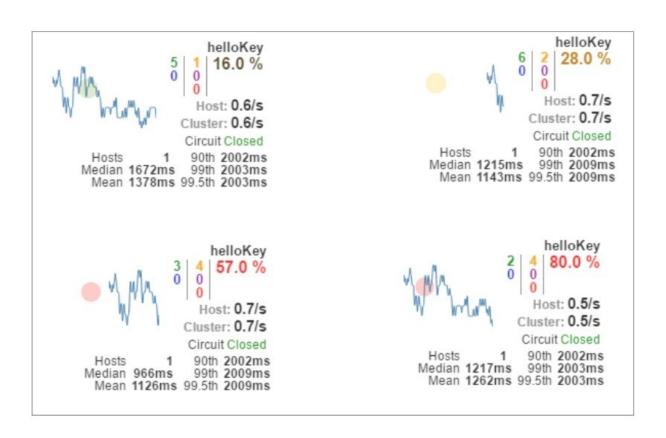
● 🛮 🖺 🗎 🗎 pom.xml 🖺 🗎 dependencies 🖺 🖺 🗎 🗎 spring-boot-

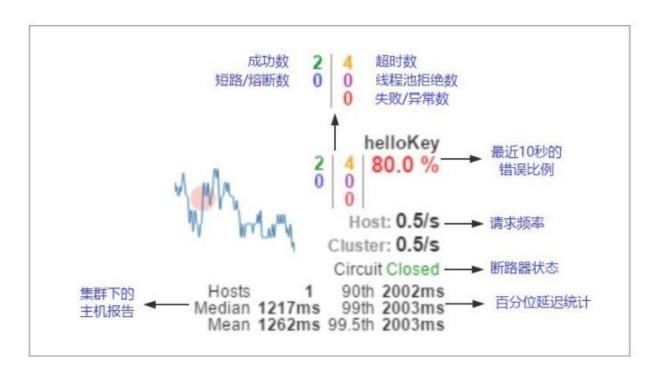
```
cloud-starter-hystrix□
    <dependencies>
       <dependency>
         <groupId>org.springframework.cloud</groupId>
         <artifactId>spring-cloud-starter-
      hystrix</artifactId>
       </dependency>
       <dependency>
         <groupId>org.springframework.boot</groupId>
         <artifactId>spring-boot-starter-
      actuator</artifactId>
       </dependency>
    </dependencies>
   ПП
   o.s.b.a.e.mvc.EndpointHandlerMapping
                                      : Mapped
   "{[/hystrix.stream/**]}" onto
    public org.springframework.web.servlet.ModelAndView
    org.springframework.cloud.netflix.endpoint.ServletWra
 ppingEndpoint.handle□javax.ser
    vlet.http.HttpServletRequest,javax.servlet.http.HttpSer
 vletResponse∏throws
    java.lang.Exception
```

| http://localhost:9000/hystrix.stream 🛮 🖶 🗎 🗎 🗎 🗎 🗘 RIBBON- |
|--|
| CONSUMER |

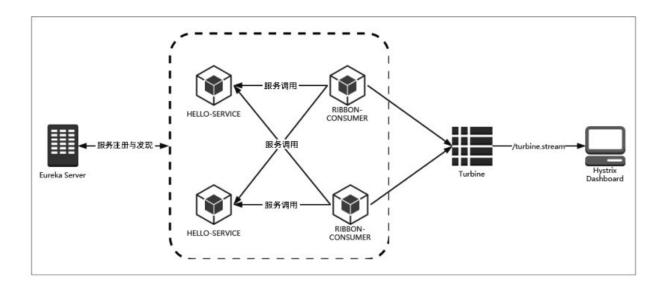


- - ullet





| Hystrix Dashboard |
|------------------------------------|
| |
| Turbine_Hystrix Dashboard |
| |
| |
| Thread Pool Loading Zuul |
| Hystrix |
| |
| |
| <u>Turbine</u> □□□ |
| |
| Hystrix Dashboard |
| /turbine.stream |
| Turbine |
| |
| |
| |
| |
| |
| |
| |
| |



- ●□□□□□□Spring Boot□□□□□turbine□
- ●□□pom.xml□□□□□□□□□□
 - properties>

project.build.sourceEncoding>UTF-

8</project.build.sourceEncoding>

oproject.reporting.outputEncoding>UTF-

8</project.reporting.outputEncoding>

<java.version>1.8</java.version>

</properties>

<parent>

<groupId>org.springframework.cloud</groupId>

<artifactId>spring-cloud-starter-

parent</artifactId>

<version>Brixton.SR5</version>

<relativePath /> <!--lookup parent from repository-

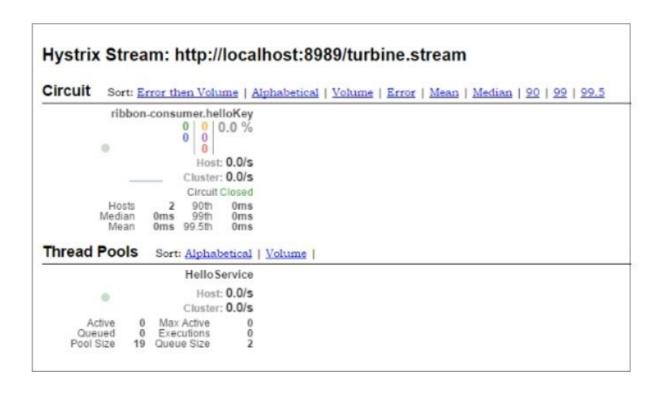
->

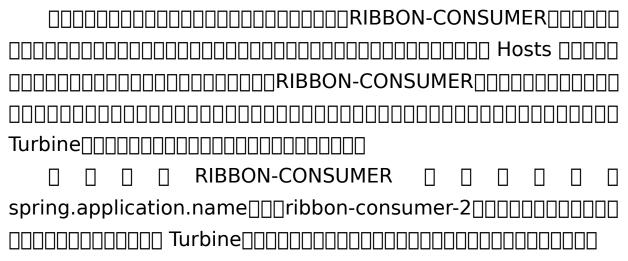
</parent>

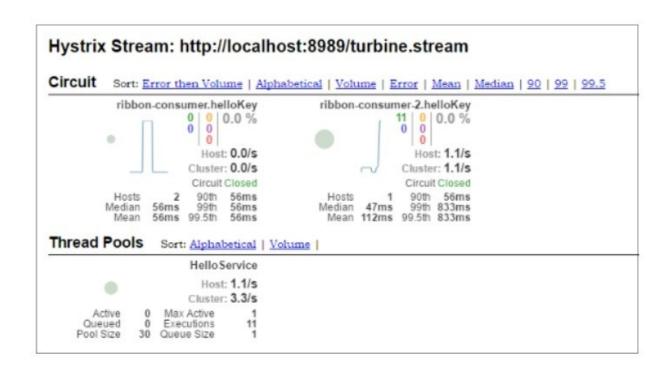
```
<dependencies>
        <dependency>
          <groupId>org.springframework.cloud
          <artifactId>spring-cloud-starter-
       turbine</artifactId>
        </dependency>
        <dependency>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-starter-
       actuator</artifactId>
        </dependency>
     </dependencies>
   ● □ □ □ □ □ □ Turbine Application □ □ □ □ @ Enable Turbine □ □ □ □
Turbine□
     @EnableTurbine
     @EnableDiscoveryClient
     @SpringBootApplication
     public class TurbineApplication {
        public static void main

☐String[]args
☐{
          SpringApplication.run
       □TurbineApplication.class,args□;
   spring.application.name=turbine
     server.port=8989
     management.port=8990
```

| eureka.client.serviceUrl.defaultZone=http://localhost:1 |
|--|
| 111/eureka/ |
| turbine.app-config=RIBBON-CONSUMER |
| turbine.cluster-name-expression="default" |
| turbine.combine-host-port=true |
| <pre>[] [turbine.app-config [] [] [] [] [] [] [] [] [] [</pre> |
| $turbine.cluster-name-expression ~ \verb \verb \verb \verb \verb \verb \verb \verb$ |
| Turbine |
| |
| $ \begin{tabular}{lllllllllllllllllllllllllllllllllll$ |
| 00000000000000000000000000000000000000 |
| |
| Turbineeureka- |
| server \square HELLO-SERVICE \square RIBBON-CONSUMER \square Turbine \square \square |
| Hystrix Dashboard [] [] Hystrix Dashboard [] [] [] |
| http://localhost:8989/turbine.stream |

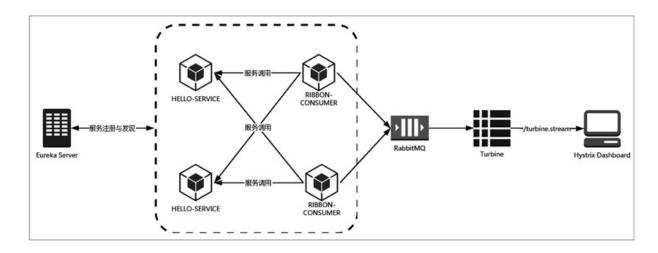






Spring Cloud

Turbine



```
●□□□□□□Spring Boot□□□□□turbine-amqp□
   properties>
      project.build.sourceEncoding>UTF-
   8</project.build.sourceEncoding>
      project.reporting.outputEncoding>UTF-
   8</project.reporting.outputEncoding>
      <java.version>1.8</java.version>
    </properties>
    <parent>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-parent</artifactId>
      <version>Brixton.SR5</version>
      <relativePath /> <!--lookup parent from repository--
   >
    </parent>
    <dependencies>
      <dependency>
       <groupId>org.springframework.cloud</groupId>
       <artifactId>spring-cloud-starter-turbine-
    amqp</artifactId>
      </dependency>
      <dependency>
       <groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-starter-
     actuator</artifactId>
      </dependency>
     </dependencies>
   □□□□□□□□□□spring-cloud-starter-turbine-amqp□□□□□□□□
      spring-cloud-starter-turbine-stream ☐
                                        pring-cloud-
____Java 8____
   ●□□□□□□□@EnableTurbineStream□□□□□Turbine Stream□
@EnableTurbineStream
     @EnableDiscoveryClient
     @SpringBootApplication
    public class TurbineApplication {
      public static void main [] args [] args [] {
        SpringApplication.run
    □TurbineApplication.class,args□;
      }
   ●□□application.properties□□□
    spring.application.name=turbine
    server.port=8989
    management.port=8990
    eureka.client.serviceUrl.defaultZone=http://localhost:1
 111/eureka/
   _____RabbitMQ______pom.xml____
```

| Spring Cloud Feign |
|---|
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Cloud Feign |
| ● |
| pom.xml [] [] spring-cloud-starter-eureka [] spring-cloud- |
| starter-feign |
| <pre><parent></parent></pre> |
| <pre><groupid>org.springframework.boot</groupid></pre> |
| <artifactid>spring-boot-starter-parent</artifactid> |
| <pre><version>1.3.7.RELEASE</version></pre> |
| <relativepath></relativepath> lookup parent from repository |
| |
| <dependencies></dependencies> |
| <dependency></dependency> |
| <pre><groupid>org.springframework.boot</groupid></pre> |
| <artifactid>spring-boot-starter-web</artifactid> |
| |
| <dependency></dependency> |
| <pre><groupid>org.springframework.cloud</groupid></pre> |
| <artifactid>spring-cloud-starter-</artifactid> |
| eureka |

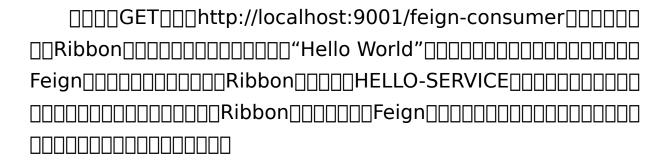
```
<dependency>
         <groupId>org.springframework.cloud</groupId>
         <artifactId>spring-cloud-starter-feign</artifactId>
       </dependency>
     </dependencies>
     <dependencyManagement>
       <dependencies>
         <dependency>
           <groupId>org.springframework.cloud</groupId>
           <artifactId>spring-cloud-
       dependencies</artifactId>
           <version>Brixton.SR5/version>
           <type>pom</type>
           <scope>import</scope>
         </dependency>
       </dependencies>
     </dependencyManagement>
   ●□□□□□□ConsumerApplication□□□□@EnableFeignClients□
□□□Spring Cloud Feign□□□□□
     @EnableFeignClients
     @EnableDiscoveryClient
     @SpringBootApplication
     public class ConsumerApplication {
       public static void main [] String[] args [] {
         SpringApplication.run
     □ConsumerApplication.class,args□;
```

</dependency>

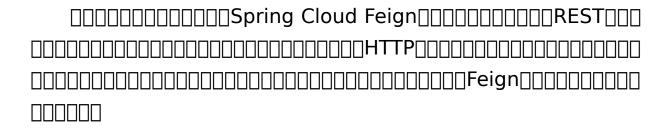
```
}
  Spring MVC
   @FeignClient[]"hello-service"[]
   public interface HelloService {
     @RequestMapping[]"/hello"[]
     String hello□□;
   }
  חחחחחחח Brixton.SR5חחחחחח serviceld
□□□□□□name□value□
  @RestController
   public class ConsumerController {
    @Autowired
    HelloService helloService:
    @RequestMapping
                         value="/feign-
  consumer",method=RequestMethod.GET[]
    public String helloConsumer∏∏{
     return helloService.hello∏;
    }
  ● □ □ □ Ribbon □ □ □ □ □ □ □ □ □ □ □ application.properties □ □ □ □
```

| spring.application.name=feign-consumer |
|---|
| server.port=9001 |
| eureka.client.serviceUrl.defaultZone=http://localhost:1 |
| 111/eureka/ |
| |
| RibbonHELLO |
| SERVICE |
| |
| |

| Instances currently registered with Eureka | | | | |
|--|---------|--------------------|--|--|
| Application | AMIs | Availability Zones | Status | |
| FEIGN-CONSUMER | n/a (1) | (1) | UP (1) - PC-201602152056 feign-consumer:9001 | |
| HELLO-SERVICE | n/a (2) | (2) | UP (2) - PC-201602152056:hello-service:8081 , PC-201602152056:hello-service:8082 | |







```
One of the spring Cloud Feign One of

\Pi

RequestBody 
\Pi

\Pi

\Pi

                   @RequestMapping
      □value="/hello1",method=RequestMethod.GET□
                   public String hello∏@RequestParam String name∏{
                               return "Hello "+name;
                   }
                   @RequestMapping
      □value="/hello2",method=RequestMethod.GET□
                   public
                                                                                                             @RequestHeader
                                                                                                                                                                               String
                                                User
                                                                         hello
                                                                                                 П
      name,@RequestHeader Integer age∏{
                               return new User \name, age \n;
                   @RequestMapping
      □value="/hello3",method=RequestMethod.POST□
                   public String hello∏@RequestBody User user∏{
                               return "Hello "+user.getName [] + ", "+user.getAge
                   \square\square;
                   }
            ____Spring Cloud Feign__JSON____User____
                   public class User {
                               private String name;
                               private Integer age;
                               public User∏∏{
```

```
public User String name, Integer age {
                                         this.name=name;
                                         this.age=age;
                                  }
                                  //□□getter□setter
                                  @Override
                                  public String toString□□{
                                         return "name="+name+",age="+age;
                                  }
                     }
              nello-service nellong hello-service nellong nellong nellong hello-service nellong nell
consumer
              • חחחח HelloService חחחחחחחחחחחחחחחחחחחחחחחחחחח
HelloService∏∏∏∏∏∏
                    @FeignClient∏"HELLO-SERVICE"∏
                    public interface HelloService {
                                  @RequestMapping[]"/hello"[]
                                  String hello∏;
                                  @RequestMapping
                    □value="/hello1",method=RequestMethod.GET□
                                  String hello □ @RequestParam □ "name" □ String
                     name∏;
                                  @RequestMapping
                    □value="/hello2",method=RequestMethod.GET□
                                  User hello [] @RequestHeader [] "name" [] String
                    name,@RequestHeader∏"age"∏Integer age∏;
```

```
@RequestMapping
    □value="/hello3",method=RequestMethod.POST□
       String hello∏@RequestBody User user∏;
    }
   nnnnnnnnnnnnnnvaluennnnnnSpring MVCnnnnnnnnnnnn
_____Feign_____value_______
java.lang.lllegalStateException:
    Caused
               by:
 RequestParam.value was empty on parameter 0
      at feign.Util.checkState∏Util.java:128∏
   ●□□□□ConsumerController□□□□□/feign-consumer2□□□□□□
@RestController
    public class ConsumerController {
      @Autowired
      HelloService helloService:
                                    value="/feign-
      @RequestMapping
   consumer",method=RequestMethod.GET[]
      public String helloConsumer∏∏{
       return helloService.hello∏;
      @RequestMapping
                                    value="/feign-
   consumer2",method=RequestMethod.GET

      public String helloConsumer2□□{
       StringBuilder sb=new StringBuilder□□;
       sb.append[helloService.hello[].append["\n"];
```

```
sb.append | helloService.hello | "DIDI" | .append
  ∏"\n"∏;
    sb.append | helloService.hello | "DIDI", 30 | | .append
  ∏"\n"∏;
    sb.append ☐ helloService.hello ☐
                          User
                       new
  \square"DIDI",30\square\square.append\square"\n"\square;
    return sb.toString∏;
   }
  }
 ____hello-service
Hello World
  Hello DIDI
  name=DIDI,age=30
  Hello DIDI,30
 Spring Cloud Feign
```

```
OCCUPATION OF THE PROPERTY OF 
● ПППППП DTO ППППППППППППППП Maven ППППППhello-
service-api∏
            <groupId>com.didispace
                  <artifactId>hello-service-api</artifactId>
                  <version>0.0.1-SNAPSHOT
                  <packaging>jar</packaging>
                  <name>hello-service-api</name>
                  <parent>
                        <groupId>org.springframework.boot</groupId>
                        <artifactId>spring-boot-starter-parent</artifactId>
                        <version>1.3.7.RELEASE
                        <relativePath/> <!--lookup parent from repository-->
                  </parent>
                  properties>
                        project.build.sourceEncoding>UTF-
            8</project.build.sourceEncoding>
                        project.reporting.outputEncoding>UTF-
            8</project.reporting.outputEncoding>
                        <java.version>1.8</java.version>
                  </properties>
                  <dependencies>
                        <dependency>
```

```
<groupId>org.springframework.boot</groupId>
       <artifactId>spring-boot-starter-web</artifactId>
     </dependency>
    </dependencies>
  com.didispace.dto.User

☐
           hello-service-api
                         П
                                     П
com.didispace.service.HelloService
□□□com.didispace.dto.User□
  ● □ □ □ □ □ □ □ □ □ □ □ com.didispace.dto.User □ □ □
□□□□com.didispace.dto.User□
    @RequestMapping[]"/refactor"[]
    public interface HelloService {
     @RequestMapping
  □value="/hello4",method=RequestMethod.GET□
     String hello□@RequestParam□"name"□String name□;
     @RequestMapping
  □value="/hello5",method=RequestMethod.GET□
     User hello ☐ @RequestHeader ☐ "name" ☐ String
  name,@RequestHeader[]"age"[Integer age[];
     @RequestMapping
  □value="/hello6",method=RequestMethod.POST□
     String hello@RequestBody User user[];
    }
```

```
□□□□/hello4□/hello5□/hello6□
   ●□□□ hello-service □□□□□□ pom.xml □ dependency □□□□□
□□hello-service-api□□□□□
     <dependency>
       <groupId>com.didispace
       <artifactId>hello-service-api</artifactId>
       <version>0.0.1-SNAPSHOT</version>
     </dependency>
   ■□□ RefactorHelloController □□□ hello-service-api □□□□
@RestController
             class RefactorHelloController
     public
                                           implements
 HelloService {
       @Override
       public String hello | @RequestParam | "name" | String
   name∏{
        return "Hello "+name;
       }
       @Override
       public User hello | @RequestHeader | "name" | String
   name,@RequestHeader[]"age"[]Integer age[]{
        return new User \name, age \n;
       }
       @Override
       public String hello∏@RequestBody User user∏{
        return "Hello "+user.getName∏ +","+user.getAge
     \Box\Box;
```

```
}
  ●□□□□□□□□□□□□□□□□□□□ feign-consumer □ pom.xml□□
____hello-service-api
  ■□□ RefactorHelloService □□□□□□ hello-service-api □□□
HelloService
   public
          interface
                  RefactorHelloService
                                  extends
 com.didispace.service.HelloService {
    }
  @Autowired
   RefactorHelloService refactorHelloService:
                              value="/feign-
   @RequestMapping
 consumer3",method=RequestMethod.GET

   public String helloConsumer3∏{
     StringBuilder sb=new StringBuilder□□;
     sb.append
                       refactorHelloService.hello
  \square"MIMI"\square.append\square"\n"\square;
     sb.append
                       refactorHelloService.hello
                 \square"MIMI",20\square.append\square"\n"\square;
     sb.append
             □ refactorHelloService.hello
                                     new
  com.didispace.dto.User[]"MIMI",20[][].append[]"\n"[];
```



<u>Ribbon</u>∏

| Spring Cloud FeignSpring | g Cloud Ribbon∏[|
|-------------------------------------|------------------|
| | |
| □Spring Cloud Feign□□□□□Ribbon□□□□□ | |

ribbon.ConnectTimeout=500 ribbon.ReadTimeout=5000

| Spring Cloud Feign |
|--|
| 🛮 🖟 Spring Cloud Ribbon 🖺 🖺 🗎 🗎 🗎 🗎 🗎 🖺 |
| <client>.ribbon.key=value </client> |
| <client>[][][]Ribbon[][][][][]</client> |
| Feign@FeignClient |
| Spring Cloud Feign |
| Ribbon |
| _value="HELLO-SERVICE" Feign |
| HELLO-SERVICE Ribbon Reign Ribbon Reign Re |
| name value Ribbon |

HELLO-SERVICE.ribbon.ConnectTimeout=500

HELLO-SERVICE.ribbon.ReadTimeout=2000

HELLO-SERVICE.ribbon.OkToRetryOnAllOperations=true

HELLO-SERVICE.ribbon.MaxAutoRetriesNextServer=2

HELLO-SERVICE.ribbon.MaxAutoRetries=1



| Spring Cloud Feign DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
|--|
| SERVICE4 |
| Spring Cloud Ribbon |
| ● hello-service |
| @RequestMapping |
| <pre>[value="/hello",method=RequestMethod.GET[]</pre> |
| public String hello□□throws Exception { |
| ServiceInstance |
| instance=client.getLocalServiceInstance[[]; |
| |
| int sleepTime=new Random ☐.nextInt ☐3000 ☐; |
| logger.info[]"sleepTime:"+sleepTime[]; |
| Thread.sleep[sleepTime[]; |
| logger.info "/hello,host:"+instance.getHost |
| +",service_id:"+ |
| instance.getServiceId□□□; |
| return "Hello World"; |
| } |
| ● [] feign-consumer [] [] [] [] [] [] [] [] [] [] [] [] [] |
| HELLOSERVICE.ribbon.MaxAutoRetries |
| HELLO- |
| SERVICE.ribbon.MaxAutoRetriesNextServer |
| |
| |
| http://localhost:9001/feignconsumer |
| hello-service |
| |

INFO 19264---[nio-8001-exec-6]com.didispace.web.HelloController :sleepTime:2929 19264---[nio-8001-exec-INFO 7]com.didispace.web.HelloController : sleepTime:253 **INFO** 19264---[nio-8001-exec-7]com.didispace.web.HelloController : /hello,host: 192.168.0.105, service id:hello-service INFO 19264---[nio-8001-exec-6]com.didispace.web.HelloController : /hello,host: 192.168.0.105, service id:hello-service On the state of th

Hystrix[[

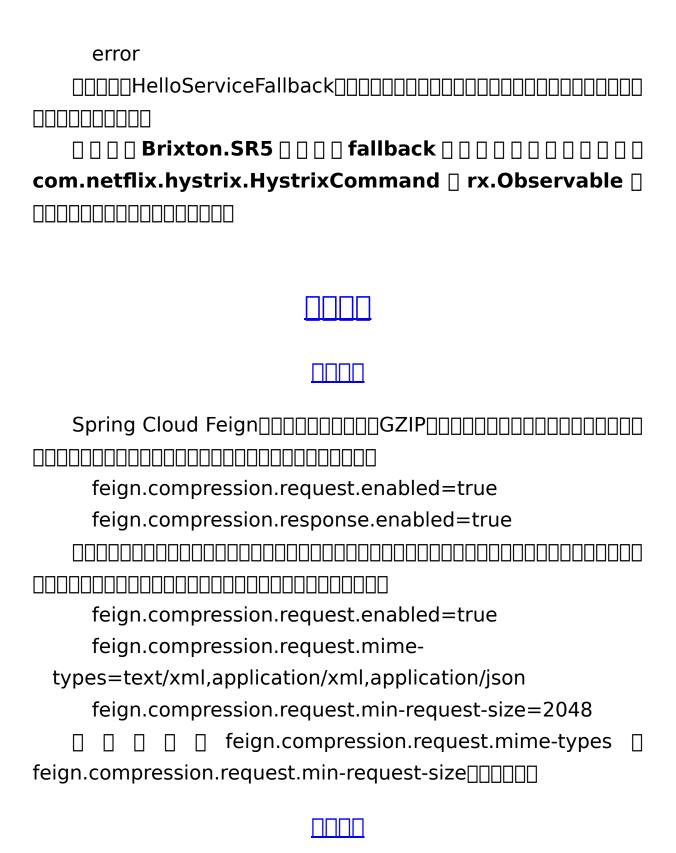
Spring Cloud Feign Cloud Ribbon Cloud Ribbon Cloud Feign Cloud Clo

| ■ HelloService @FeignClient |
|---|
| |
| @FeignClient 🛮 name="HELLO- |
| ${\sf SERVICE",} configuration = {\sf DisableHystrixConfiguration.} class$ |
| |
| public interface HelloService { |
| ••• |
| } |
| |
| |
| Hystrix |
| Hystrix hystrix.command. |
| <commandkey>[][][][]<commandkey>[][][][][][]Feign[][][]</commandkey></commandkey> |
| |
| <pre> </pre> |
| hystrix.command.hello.execution.isolation.thread.time |
| outInMilliseconds=5000 |
| Hystrix |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| |
| Hystrix |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| <pre></pre> |

```
fallback
_____feign-consumer_____
  @Component
          class HelloServiceFallback
     public
                                 implements
  HelloService {
      @Override
      public String hello□□{
        return "error":
      }
      @Override
      public String hello | @RequestParam | "name" | String
    name∏{
        return "error";
      }
      @Override
      public User hello | @RequestHeader | "name" | String
    name,@RequestHeader[]"age"[]
    Integer age<sub>□</sub>{
        return new User∏"∏∏",0∏;
      @Override
      public String hello∏@RequestBody User user∏{
        return "error":
      }
```

```
● □ □ □ □ □ □ HelloService □ □ □ @ FeignClient □ □ □ fallback □ □ □ □
@FeignClient
                                       name="HELLO-
                           П
 SERVICE",fallback=HelloServiceFallback.class

     public interface HelloService {
       @RequestMapping[]"/hello"[]
      String hello∏∏;
       @RequestMapping
   □value="/hello1",method=RequestMethod.GET□
       String hello□@RequestParam□"name"□String name□;
       @RequestMapping
   □value="/hello2",method=RequestMethod.GET□
       User hello ☐ @RequestHeader ☐ "name" ☐ String
   name,@RequestHeader∏"age"∏Integer age∏;
       @RequestMapping
   □value="/hello3",method=RequestMethod.POST□
      String hello∏@RequestBody User user∏;
     }
   hello-service [ [ ] [ ] GET [ ] [ ] http://localhost:9001/feign-
consumer2 | | | | | | | HelloService | | | 4 | | | | | | | | | | hello-
service
     error
     error
     name = \prod \bigcap age = 0
```

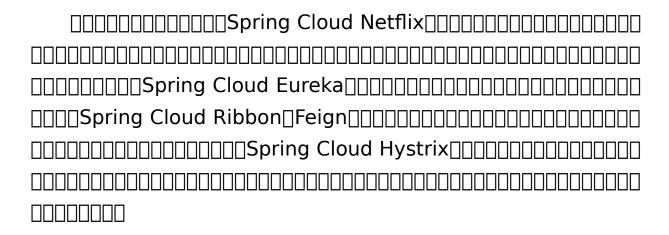


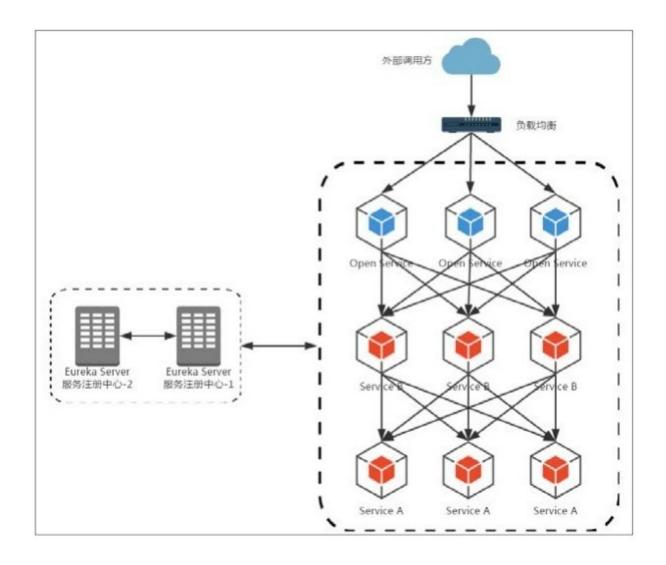
```
HelloService⊓⊓⊓⊓⊓⊓⊓⊓
  logging.level.com.didispace.web.HelloService=DEBUG
 Logger.Level
_____Logger.Level_Bean____
@EnableFeignClients
  @EnableDiscoveryClient
  @SpringBootApplication
  public class ConsumerApplication {
   @Bean
   Logger.Level feignLoggerLevel∏{{
    return Logger.Level.FULL;
   }
   public static void main \ String[]args \ {
    SpringApplication.run
  □ConsumerApplication.class,args□;
   }
 nnnnnnnnnnnnn Feign nnnnnnnnnnnnnnnnnnnn
@Configuration
```

```
DEBUG 20488 --- [HELLO-SERVICE-1] com.didispace.web.HelloService:
[HelloService#hello] ---> GET http://HELLO-SERVICE/hello HTTP/1.1
    DEBUG 20488 --- [HELLO-SERVICE-1] com.didispace.web.HelloService :
[HelloService#hello] ---> END HTTP (0-byte body)
    DEBUG 20488 --- [HELLO-SERVICE-1] com.didispace.web.HelloService :
[HelloService#hello] <--- HTTP/1.1 200 OK (3342ms)
    DEBUG 20488 --- [HELLO-SERVICE-1] com.didispace.web.HelloService:
[HelloService#hello] Server: Apache-Coyote/1.1
    DEBUG 20488 --- [HELLO-SERVICE-1] com.didispace.web.HelloService:
[HelloService#hello] Content-Length: 11
    DEBUG 20488 --- [HELLO-SERVICE-1] com.didispace.web.HelloService :
[HelloService#hello] X-Application-Context: hello-service:8001
    DEBUG 20488 --- [HELLO-SERVICE-1] com.didispace.web.HelloService :
[HelloService#hello] Date: Sat, 07 Jan 2017 11:49:11 GMT
    DEBUG 20488 --- [HELLO-SERVICE-1] com.didispace.web.HelloService:
[HelloService#hello] Content-Type: text/plain; charset=UTF-8
    DEBUG 20488 --- [HELLO-SERVICE-1] com.didispace.web.HelloService:
[HelloService#hello]
    DEBUG 20488 --- [HELLO-SERVICE-1] com.didispace.web.HelloService :
[HelloService#hello] Hello World
    DEBUG 20488 --- [HELLO-SERVICE-1] com.didispace.web.HelloService:
[HelloService#hello] <--- END HTTP (11-byte body)
```

| | ∏Feig | $n \square I \cap i$ | agar | | | | | ПΠ | | ППГ | | $\neg \sqcap$ | ПΓ | $\neg \vdash$ | П | П |
|---|----------|----------------------|-------|---------|--|---------------|--------|--------|-----|-------|-------|---------------|----|---------------|------|---|
| П | ı ıı eig | | JUELL | II II I | | 4 | 1 11 1 | 1 II I | III | II II | II II | Ш | ΙШ | Ш | II I | П |

□7 API□□□□Spring Cloud Zuul





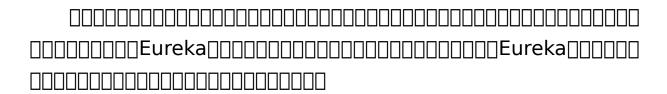
| APISpring Cloud |
|---|
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Cloud Zuul |
| DODDODDODDODDODDODDODDODDODDODDODDODDOD |
| Cloud Eureka |
| |
| |
| |
| |
| |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| |

| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
|--|
| |
| |
| |
| |
| |
| |
| |
| Spring Cloud Zuul |
| |
| |
| |
| |
| |
| |
| Spring Cloud Zuul |
| |
| |
| |
| |
| $hello-service \verb feign-consumer \verb feign-consumer $ |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| consumer [|
| http://localhost:9001/feign-consumer |
| |

```
●□□□□□□□Spring Boot□□□□□□api-gateway□□□pom.xml□□□
spring-cloud-starter-zuul
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.3.7.RELEASE/version>
        <relativePath/>
    </parent>
    <dependencies>
        <dependency>
         <groupId>org.springframework.cloud</groupId>
         <artifactId>spring-cloud-starter-
      zuul</artifactId>
        </dependency>
    </dependencies>
    <dependencyManagement>
        <dependencies>
         <dependency>
            <groupId>org.springframework.cloud
         pld>
            <artifactId>spring-cloud-
         dependencies</artifactId>
            <version>Brixton.SR5</version>
            <type>pom</type>
            <scope>import</scope>
```

```
</dependency>
                            </dependencies>
                 </dependencyManagement>
           One of the control of
■spring-boot-starter-actuator
Spring Cloud Zuulnnnnnn/routesnnnnnnnnnnnnn
           ●□□□□□□□□@EnableZuulProxy□□□□Zuul□API□□□□□□□
                 @EnableZuulProxy
                @SpringCloudApplication
                public class Application {
                      public static void main [] args [] args [] {
                                                                                                       SpringApplicationBuilder
                            new
                □Application.class□.web□true□.run□args□;
                      }
                 }
           spring.application.name=api-gateway
                server.port=5555
```





| Instances currently registered with Eureka | | | |
|--|---------|--------------------|--|
| Application | AMIs | Availability Zones | Status |
| FEIGN-CONSUMER | n/a (1) | (1) | UP (1) - PC-201602152056:feign-consumer:9001 |
| HELLO-SERVICE | n/a (1) | (1) | UP (1) - PC-201602152056:hello-service |

| Spring Cloud Zuul api-gateway |
|--|
| |
| zuul.routes.api-a-url.path=/api-a-url/** |
| zuul.routes.api-a-url.url=http://localhost:8080/ |
| API/api-a-url/** |
| http://localhost:8080/ |
| http://localhost:5555/api-a-url/hello |
| http://localhost:8080/hello |
| zuul.routes.api-a-url.path 🛮 api-a-url 🔻 🔻 🔻 🔻 🔻 🔻 🔻 🔻 🔻 🔻 |
| path[]url |
| |
| |

| ● 🛮 🗎 🗎 Eureka 🔻 🖺 🖺 🖺 🖺 🗎 api-gateway 🗎 pom.xml |
|---|
| springcloud-starter-eureka |
| <dependency></dependency> |
| <pre><groupid>org.springframework.cloud</groupid></pre> |
| <artifactid>spring-cloud-starter-eureka</artifactid> |
| |
| ■□api-gateway□application.properties□□□□□□□□□Eureka□□□ |
| |
| zuul.routes.api-a.path=/api-a/** |
| zuul.routes.api-a.serviceld=hello-service |
| zuul.routes.api-b.path=/api-b/** |
| zuul.routes.api-b.serviceld=feign-consumer |
| eureka.client.serviceUrl.defaultZone=http://localhost:1 |
| 111/eureka/ |
| |
| api-a _ api-bEureka Server |
| $\verb $ |
| $consumer \verb $ |
| |
| |
| $feign-consumer \verb $ |
| <pre>Documentation</pre> |
| feign-consumer[][][][][][][API-GATEWAY[] |

| Instances currently registered with Eureka | | | | | | |
|--|---------|--------------------|--|--|--|--|
| Application | AMIs | Availability Zones | Status | | | |
| API-GATEWAY | n/a (1) | (1) | UP (1) - PC-201602152056:api-gateway:5555 | | | |
| FEIGN-CONSUMER | n/a (1) | (1) | UP (1) - PC-201602152056:feign-consumer:9001 | | | |
| HELLO-SERVICE | n/a (1) | (1) | UP (1) - PC-201602152056:hello-service | | | |



```
ON APICO OF THE STATE OF THE ST
____ZuulFilter_____4____4_____
             ______________Zuul ________________
□□401 Unauthorized□□□
                          public class AccessFilter extends ZuulFilter {
                                private static Logger log=LoggerFactory.getLogger
                   ∏AccessFilter.class∏;
                                @Override
                                public String filterType∏{
                                       return "pre";
                                 }
                                @Override
                                public int filterOrder□□{
                                       return 0;
                                 }
                                @Override
                                public boolean shouldFilter∏∏{
                                       return true:
                                 }
```

```
@Override
      public Object run□□{
       RequestContext
     ctx=RequestContext.getCurrentContext[];
       HttpServletRequest request=ctx.getRequest□□;
       log.info
                         {}
                  "send
                              request
                                      to
     {}",request.getMethod□□,
   accessToken=request.getParameter
       Object
     □"accessToken"□;
       if∏accessToken==null∏{
         log.warn∏"access token is empty"∏;
        ctx.setSendZuulResponse[]false[];
        ctx.setResponseStatusCode∏401∏;
         return null:
       log.info□"access token ok"□;
       return null:
       }
      }
  •filterOrder_______
```



| ullet |
|---|
| ullet |
| ullet |
| ullet |
| |
| Spring Cloud ZuulAPI |
| |
| |
| |
| |
| |
| Spring Cloud Zuul |
| |
| |
| |
| |
| |
| • [] [] zuul.routes. <route>.path [] zuul.routes.</route> |
| <route>.url</route> |
| zuul.routes.user-service.path=/user-service/** |
| zuul.routes.user-service.url=http://localhost:8080/ |

| http://localhost:8080/ |
|---|
| http://localhost:5555/user-service/hello 🛮 🗎 🗎 🔻 API |
| /user-service/hello path API |
| http://localhost:8080/hello |
| ● 🛮 🗎 🗎 🗎 🗎 zuul.routes. <route>.path 🗎 zuul.routes.</route> |
| <route>.serviceId</route> |
| zuul.routes.user-service.path=/user-service/** |
| zuul.routes.user-service.serviceld=user-service |
| ribbon.eureka.enabled=false |
| user- |
| service.ribbon.listOfServers=http://localhost:8080/,http://lo |
| calhost:8081/ |
| |
| $http://localhost:8080/ \verb http://localhost:8081/ \verb \verb \verb \verb \verb \verb \verb \verb \verb \verb $ |
| zuul.routes. <route>.path _</route> |
| $zuul.routes. < route > .service Id \verb $ |
| ribbon.listOfServers |
| API |
| DDDDSpring Cloud Zuul |
| |
| ■ ribbon.eureka.enabled 🛮 🖺 zuul.routes. |
| <route>.serviceId [][][][][][][][][][][][Ribbon [][][][][][][][][][][][][][][][][][][]</route> |
| 00000000000000000000000000000000000000 |
| false serviceId |
| ■ user-service.ribbon.listOfServers [] [] [] [] zuul.routes. |
| <route>.serviceId [][][][][] user-service [][][]serviceId[][][]</route> |

| <route>[][][] < route > [][][] path [][][][][][][][][][][][][][][][][][][]</route> |
|---|
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| |
| |
| Spring Cloud Zuul |
| Spring Cloud Eureka |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| <route>.path[]zuul.routes.<route>.serviceId[][][][][][][][][]</route></route> |
| |
| service |
| zuul.routes.user-service.path=/user-service/** |
| zuul.routes.user-service.serviceld=user-service |
| path_serviceId |
| $\verb zuul.routes. < serviceld > = < path > $ |
| path>path |
| serviceId |
| zuul.routes.user-service=/user-service/** |
| UAPIURLpath |
| urlserviceId |
| $\verb path service d $ |
| |
| 00000000000 Zuul 00000 Eureka 00000000000000 |
| APIEurekaEureka_ |

| DDDDDDDDDDDDDServiceIdDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
|---|
| URLpathAPI |
| serviceId |
| Ribbon |
| |
| |
| Eureka_Zuul |
| |
| path user- |
| service[][][][][][][]user-service[] |
| zuul.routes.user-service.path=/user-service/** |
| zuul.routes.user-service.serviceld=user-service |
| 000000000000000000000000000000000Zuul0000000 |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Eureka |
| serviceId |
| Eureka Zuul |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| services DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| 00000000000000000000000000000000000000 |
| DDDDzuul.ignored-services=*DDDZuul |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| <pre>Document of the content of the</pre> |
| EurekaZuul |
| |

| <pre>[][][][][][][][][][][][][][][][][][][]</pre> |
|--|
| $orderservice-v2 \verb $ |
| <pre>Domain</pre> |
| |
| |
| $\verb $ |
| |
| userservice-v1 |
| $\verb 0000-00000000000000000000000000000000$ |
| /v1/userservice/** |
| Bean |
| @Bean |
| public PatternServiceRouteMapper serviceRouteMapper |
| |
| return new PatternServiceRouteMapper[] |
| "[]? <name>^.+[]-[]? <version>v.+\$[]",</version></name> |
| "\${version}/\${name}"[]; |
| } |
| PatternServiceRouteMapper |
| |
| API |
| PatternServiceRouteMapper |
| |

| | 1000000000 | |
|--|------------|--|
| | | |

| 通配符 | 说明 |
|-----|------------------|
| ? | 匹配任意单个字符 |
| * | 匹配任意数量的字符 |
| ** | 匹配任意数量的字符,支持多级目录 |

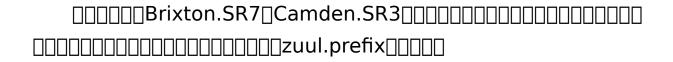
| URL 路径 | 说明 | | | |
|------------------|---|--|--|--|
| /user-service/? | 它可以匹配/user-service/之后拼接一个任务字符的路径,比如 | | | |
| | /user-service/a、/user-service/b、/user-service/c | | | |
| /user-service/* | 它可以匹配/user-service/之后拼接任意字符的路径,比如 | | | |
| | /user-service/a、/user-service/aaa、/user-service/bbb。但 | | | |
| | 是它无法匹配/user-service/a/b | | | |
| /user-service/** | 它可以匹配/user-service/*包含的内容之外,还可以匹配形如 | | | |
| | /user-service/a/b的多级目录路径 | | | |

```
zuul.routes.user-service.path=/user-service/**
   zuul.routes.user-service.serviceld=user-service
   zuul.routes.user-service-ext.path=/user-service/ext/**
   zuul.routes.user-service-ext.serviceld=user-service-ext
         user-service-ext ППП URL ПППППППП
  ппппп
@Override
   public Route getMatchingRoute∏final String path∏{
     ZuulRoute route=null;
     if□! matchesIgnoredPatterns□adjustedPath□□{
       for
             Entry<String,ZuulRoute>
                              entry
    this.routes.get∏∏.entrySet∏∏∏{
         String pattern=entry.getKey□□;
         log.debug∏"Matching pattern:"+pattern∏;
         if
                      this.pathMatcher.match
       □pattern,adjustedPath□□{
          route=entry.getValue∏;
          break:
         }
```

```
}
       log.debug∏"route matched="+route∏;
       return getRoute \( \text{route,adjustedPath} \( \text{\text{}}; \)
    }
  LinkedHashMap
protected Map<String,ZuulRoute> locateRoutes□□{
       LinkedHashMap<String,ZuulRoute>
    routesMap=new LinkedHashMap<String,ZuulRoute>
    \Pi\Pi;
      for \( \text{ZuulRoute} \) route : this.properties.getRoutes
    ∏∏.values∏∏∏{
        routesMap.put□route.getPath□□,route□;
       }
       return routesMap;
    }
  zuul:
     routes:
       user-service-ext:
        path: /user-service/ext/**
        serviceld: user-service-ext
       user-service:
        path: /user-service/**
```

serviceld: user-service ____Zuul _____zuul.ignored-patterns nnnnnnnnnnnnnn/hello nnnnnnnnnnnnnn zuul.ignored-patterns=/**/hello/** zuul.routes.api-a.path=/api-a/** zuul.routes.api-a.serviceld=hello-service hello-service □ /hello □ □ http://localhost:5555/api-a/hello | | | | | | | | | | | | | | path | | | | | | o.s.c.n.z.f.pre.PreDecorationFilter: No route found for uri: /api-a/hello ____Zuul___zuul.prefix_______ ____zuul.prefix______Bug____Bug_____

| zuul.prefix=/api |]path/api_[| |
|--|--|---|
| | | |
| zuul.routes.api-a.path=/ | api/a/** | |
| zuul.routes.api-a.service | eld=hello-service | |
| zuul.routes.api-b.path=/ | api-b/** | |
| zuul.routes.api-b.service | eld=hello-service | |
| zuul.routes.api-c.path=/ | ccc/** | |
| zuul.routes.api-c.service | ld=hello-service | |
| /api/a/**_/a | pi-b/**[]/ccc/**[][][][| |
| hello-service | ul.prefix=/api@@@@@ | |
| zuul.prefix=/api | | |
| o.s.c.n.zuul.web.ZuulF | landlerMapping | : Mapped |
| URL path[/api/api/a/a/**]o | nto | |
| · | | |
| handler | of | type[class |
| handler
org.springframework.cloud.r | of | |
| | of
netflix.zuul.web.Zuul | |
| org.springframework.cloud.r | of
netflix.zuul.web.Zuul
landlerMapping | Controller] |
| org.springframework.cloud.r | of
netflix.zuul.web.Zuul
landlerMapping | Controller] |
| org.springframework.cloud.r
o.s.c.n.zuul.web.ZuulF
URL path[/api/api-b-b/**]o | of
netflix.zuul.web.Zuul
landlerMapping
nto
of | Controller] : Mapped type[class |
| org.springframework.cloud.r
o.s.c.n.zuul.web.Zuull
URL path[/api/api-b-b/**]o
handler | of
netflix.zuul.web.Zuul
landlerMapping
nto
of
netflix.zuul.web.Zuul | Controller] : Mapped type[class |
| org.springframework.cloud.r
o.s.c.n.zuul.web.Zuull
URL path[/api/api-b-b/**]o
handler
org.springframework.cloud.r | of
netflix.zuul.web.Zuul
landlerMapping
nto
of
netflix.zuul.web.Zuul | Controller] : Mapped type[class Controller] |
| org.springframework.cloud.r
o.s.c.n.zuul.web.Zuull-
URL path[/api/api-b-b/**]o
handler
org.springframework.cloud.r
o.s.c.n.zuul.web.Zuull- | of
netflix.zuul.web.Zuul
landlerMapping
nto
of
netflix.zuul.web.Zuul | Controller] : Mapped type[class Controller] |
| org.springframework.cloud.r
o.s.c.n.zuul.web.Zuull-
URL path[/api/api-b-b/**]o
handler
org.springframework.cloud.r
o.s.c.n.zuul.web.Zuull-
URL path[/api/ccc/**]onto | of
netflix.zuul.web.Zuul
landlerMapping
nto
of
netflix.zuul.web.Zuul
landlerMapping
of | Controller] : Mapped type[class Controller] : Mapped type[class |
| org.springframework.cloud.r o.s.c.n.zuul.web.Zuull- URL path[/api/api-b-b/**]o handler org.springframework.cloud.r o.s.c.n.zuul.web.Zuull- URL path[/api/ccc/**]onto handler | of netflix.zuul.web.Zuul landlerMapping nto of netflix.zuul.web.Zuul landlerMapping of netflix.zuul.web.Zuul | Controller] : Mapped type[class Controller] : Mapped type[class Controller] |
| org.springframework.cloud.r o.s.c.n.zuul.web.Zuull- URL path[/api/api-b-b/**]o handler org.springframework.cloud.r o.s.c.n.zuul.web.Zuull- URL path[/api/ccc/**]onto handler org.springframework.cloud.r | of netflix.zuul.web.Zuul landlerMapping nto of netflix.zuul.web.Zuul landlerMapping of netflix.zuul.web.Zuul | Controller] : Mapped type[class Controller] : Mapped type[class Controller] |



```
____path_url_____url____forward______
ПП
           \label{localhost:8001/def} $$ $$ http://localhost:8001/degapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegapi-bdegap
zuul.routes.api-a.path=/api-a/**
                 zuul.routes.api-a.url=http://localhost:8001/
                 zuul.routes.api-b.path=/api-b/**
                 zuul.routes.api-b.url=forward:/local
           @RestController
                 public class HelloController {
                       @RequestMapping□"/local/hello"□
                      public String hello∏∏{
                            return "Hello World Local";
                       }
                 }
```

<u>Cookie</u>∏∏∏

| zuul.sensitiveHeaders |
|---|
| $\verb $ |
| One Cookie Spring Cloud Zuul |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Cloud Zuul |
| |
| ullet |
| zuul.sensitiveHeaders= |
| CookieAPI |
| DDDDDDDDDRESTful APIDDDDDDDDDWebDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| WebAppAPI |
| ullet |
| # 00000000000000 |
| zuul.routes. <router>.customSensitiveHeaders=true</router> |
| # 0000000000000 |
| zuul.routes. <router>.sensitiveHeaders=</router> |
| 000000000000000Web000000000000000000000 |
| |
| |
| Cookie Web |
| |
| |
| |
| |

| Spring |
|--|
| Security Shiro Shi |
| 302 Location |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Host |
| spring-cloud-netflix-core-1.2.xZuul |
| 00000000000000000000000000000000000000 |
| |
| zuul.addHostHeader=true |
| <pre>D Spring Cloud DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD</pre> |
| cloud-netflix-core-1.1.x $\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \$ |
| $netflix\text{-}core\text{-}1.2.x \verb $ |
| Brixton Camden PreDecorationFilter |
| |
| |
| <u>Hystrix</u> ∏Ribbon∏∏ |
| " " spring-cloud-starter-zuul |
| $\verb $ |
| 000000Zuul00000000000000000000000000000 |
| path_url |
| HystrixCommand DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| $\verb $ |
| |
| $\verb $ |
| |

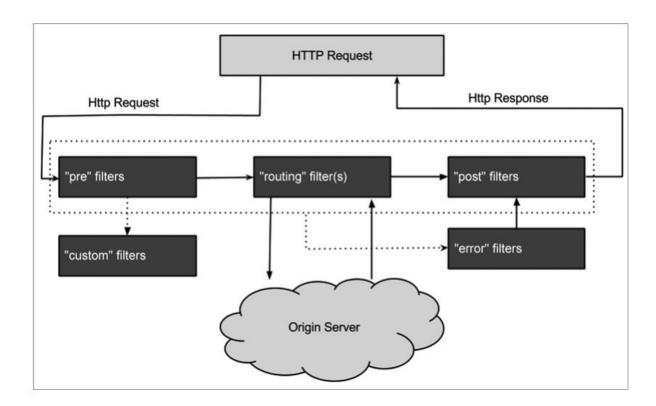
| hystrix.command.default.execution.isolation.thread.tim |
|--|
| eoutIn Milliseconds [] [] [] [] API [] [] [] [] |
| HystrixCommand |
| Hystrix DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| { |
| "timestamp": 1481350975323, |
| "status": 500, |
| "error": "Internal Server Error", |
| "exception": |
| "com.netflix.zuul.exception.ZuulException", |
| "message": "TIMEOUT" |
| } |
| ●ribbon.ConnectTimeout□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| □ □ □ ribbon.ConnectTimeout □ □ □ □ |
| hystrix.command.default.execution.isolation.thread.timeoutl |
| nMilliseconds |
| |
| { |
| "timestamp": 1481352582852, |
| "status": 500, |
| "error": "Internal Server Error", |
| "exception": |
| "com.netflix.zuul.exception.ZuulException", |
| "message": |
| "NUMBEROF_RETRIES_NEXTSERVER_EXCEEDED" |
| } |

| | | ribbon.Co | nnectTin | neout | | | | | | |
|--|--------|---|------------|----------|---------|-------|------|--------|------------------|-----|
| hystrix | .com | mand.defa | ault.exec | ution.is | solatio | on.th | read | d.time | eou [.] | tl |
| nMillise | econd | ls | | | | | | | | |
| | | | | | MEOU | T | | | | |
| ● r | ibbon | ı.ReadTim | eout 🛮 🖺 🗎 | |]000 | | | | | |
| ribbon. | Conn | ectTimeo | ut [][][| | | | | | | |
| ribbon. | Read | Timeout | | | | ĺ | | | | |
| hystrix | .com | mand.defa | ault.exec | ution.is | solatio | on.th | read | d.time | eou [.] | tl |
| nMillise | econd | s | | | | | | | | |
| | | | | | | | | Zuul | | |
| NUMBE | ROF_ | RETRIES_ | NEXTSER | RVER_E | XCEE | DED | | | | |
| ribbon. | Read | Timeout | | | | [| | | | |
| hystrix | .com | mand.defa | ault.exec | ution.is | solatio | on.th | read | d.time | eou [.] | tl |
| nMillise | econd | s | | | | | | | | |
| | | |]]]TIME(|][[[TUC | | | | | | |
| | | |][[[Zuul | | | | | | | |
| | | Hys | trix[][][[| | | | | | | |
| | | | | | | | | | | |
| Z | uul.re | etryable= | false | | | | | | | |
| Z | uul.ro | outes. <ro< td=""><td>ute>.retr</td><td>yable=</td><td>false</td><td></td><td></td><td></td><td></td><td></td></ro<> | ute>.retr | yable= | false | | | | | |
| | 🛮 zu | ul.retryab | le 🛮 🗎 🖸 | | | | Z | uul.ro | oute | es. |
| <route< td=""><td>>.ret</td><td>ryable=fa</td><td>lse[][][</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></route<> | >.ret | ryable=fa | lse[][][| | | | | | | |
| | | | | | | | | | | |
| | | | | | | | | | | |

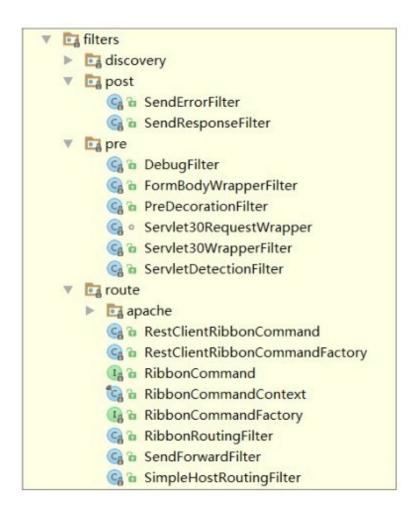


| | |][[Zuul |
|--|--|---------|
| | | |

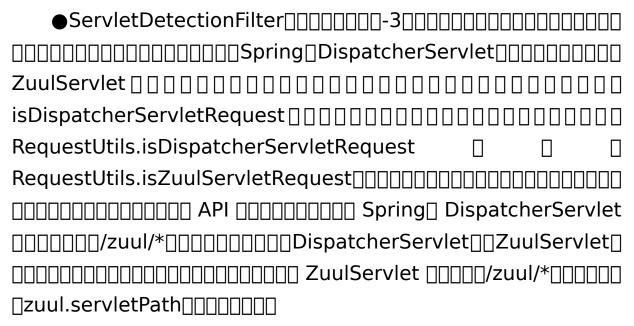
| 000000000000Zuul00000000000000000000000 |
|---|
| |
| |
| 00000000000000000000000000000000000pre00000000 |
| 00000000000000000000000000000000000000 |
| preZuulAPI |
| 000000Zuul0HTTP0000000000000000000000000000000000 |
| Spring Cloud Zuul |
| 00000000000000000000000000000000000000 |
| String filterType :: |
| int filterOrder□□; |
| boolean shouldFilter□□; |
| Object run□□; |
| |
| ●filterType□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| 000000000Zuul0000040000000000000000000 |
| ■pre |
| ■routing |
| ■ post□□routing□error□□□□□□□□□ |



| post |
|--|
| |
| post |
| |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Cloud Zuul |
| |
| |
| Spring Cloud Zuul |
| |
| spring-cloud netflix-core |
| org.springframework.cloud.netflix.zuul.filters |
| |
| Zuul |
| |

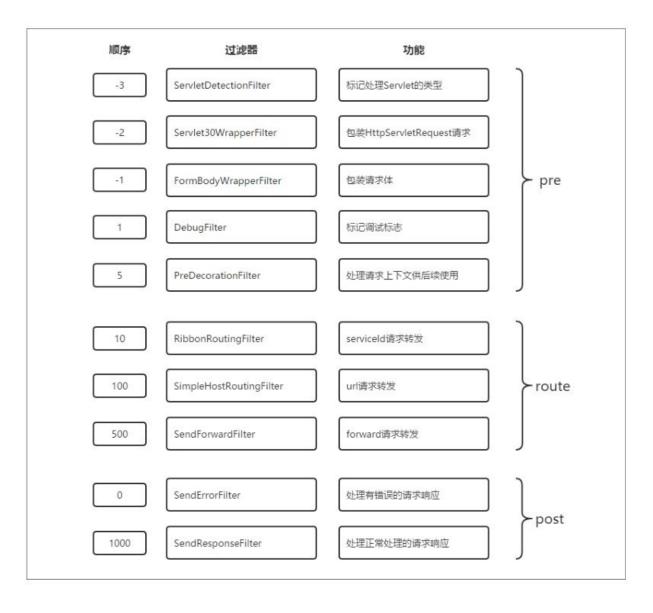


pre[[[

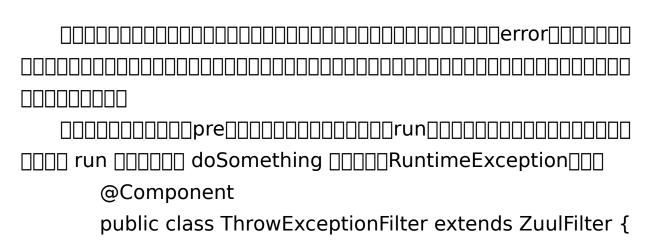


| ●Servlet30WrapperFilter□□□□□□□□-2□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
|--|
| |
| Servlet30RequestWrapper |
| ullet Form Body Wrapper Filter $ldots$ ld |
| <pre></pre> |
| urlencoded Content-Type multipart/formdata |
| Spring DispatcherServlet |
| |
| |
| ● DebugFilter |
| zuul.debug.request |
| |
| |
| debug |
| zuul.debug.parameter |
| |
| ● Pre Decoration Filter |
| forward.to_serviceId |
| |
| |
| |
| RequestContext.getCurrentContext RequestContext.getCurrentContext.getCurre |
| X- |
| Forwarded-Host 🛮 X-Forwarded-Port 🖺 🗎 🗎 🗎 🗎 🗎 🗎 🗎 🗎 🗎 |
| zuul.addProxyHeaders |
| X-Forwarded-* X-ForwardedHost X- |
| Forwarded-Port [] X-Forwarded-For [] X-Forwarded-Prefix [] X- |

| Forwarded-Proto |
|---|
| |
| route[][[|
| ullet RibbonRoutingFilter $ldots$ |
| DDDDDDDDDDDserviceIdDDDDDDDDDDDDserviceIdDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| ●SimpleHostRoutingFilter□□□□□□□100□□route□□□□□□□□ |
| urlurl |
| routeHost |
| |
| |
| ●SendForwardFilter□□□□□□□500□□route□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| forward.to |
| |
| post[][] |
| ●SendErrorFilter□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| /error |
| ●SendResponseFilter□□□□□□1000□□post□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| |
| |
| |







```
private static Logger log=
LoggerFactory.getLogger[]ThrowExceptionFilter.class[];
       @Override
       public String filterType□□{
         return "pre";
       }
       @Override
       public int filterOrder□□{
        return 0;
       }
       @Override
       public boolean shouldFilter∏∏{
        return true;
       }
       @Override
       public Object run□□{
         RuntimeException"[];
         doSomething□□;
        return null;
       }
       private void doSomething□□{
        throw new RuntimeException [] "Exist some
     errors..."□;
       }
   }
```

```
On the state of th
Spring Cloud Zuul
try-catch∏
        SendErrorFilter
_____SendErrorFilter_
shouldFilter∏∏
           public boolean shouldFilter∏∏{
                RequestContext
        ctx=RequestContext.getCurrentContext∏;
                return ctx.containsKey [] "error.status_code" [] &&!
        ctx.getBoolean[]SEND ERROR FILTER RAN,false[];
            }
        "error.status_code"
ThrowExceptionFilter
SendErrorFilter
run∏∏∏∏∏∏
           public Object run□□{
```

```
RequestContext
     context=RequestContext.getCurrentContext□□;
        this.helper.addIgnoredHeaders∏;
        try {
          RibbonCommandContext
      commandContext=buildCommandContext□context□;
         ClientHttpResponse
                                   response=forward
      \squarecommandContext\square;
         setResponse∏response∏;
         return response;
        }
        context.set
      □ERROR STATUS CODE,ex.nStatusCode□;
          context.set[]"error.message",ex.errorCause[];
         context.set□"error.exception",ex□;
        }
        catch□Exception ex□{
          context.set
      □"error.status code",HttpServletResponse.SC INTERN
      AL SERVER ERROR□;
         context.set□"error.exception",ex□;
        return null;
     }
   ______catch_____catch____catch____
```

```
●error.status code□□□□□□
   ●error.exception:Exception□□□□□
   ____ThrowExceptionFilter_run______
    public Object run□□{
       log.info 

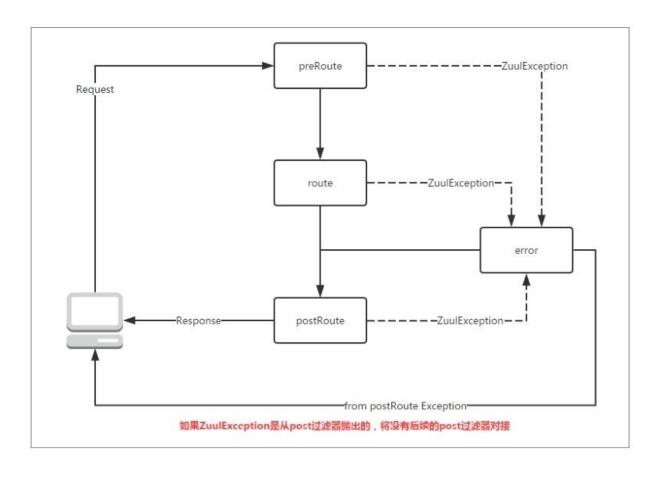
□ "This is a pre filter, it will throw a
    RuntimeException"□;
       RequestContext
    ctx=RequestContext.getCurrentContext□□;
       try {
        doSomething \square \square;
       } catch∏Exception e∏{
        ctx.set
     □"error.status code",HttpServletResponse.SC INTERN
     AL SERVER ERROR∏;
        ctx.set[]"error.exception",e[];
       }
       return null;
    }
   {
       "timestamp": 1481674980376,
       "status": 500,
       "error": "Internal Server Error".
       "exception": "java.lang.RuntimeException",
```

```
"message": "Exist some errors..."
  }
 ●status□□□error.status code□□□□□□
 ●exception□□□error.exception□□□Exception□□□□
 message[][]error.exception[][]Exception[]message[][]
ErrorFilter \square
 SendErrorFilter
_____ error ______ pre_route_
post
___ try-catch ______error _______
public class ErrorFilter extends ZuulFilter {
          log=LoggerFactory.getLogger
  Logger
 ∏ErrorFilter.class∏;
  @Override
```

```
public String filterType□□{
        return "error";
       }
       @Override
       public int filterOrder□□{
        return 10;
       }
       @Override
       public boolean shouldFilter∏∏{
        return true;
       }
       @Override
       public Object run□□{
        RequestContext
     ctx=RequestContext.getCurrentContext□□;
        Throwable throwable=ctx.getThrowable \square;
                   П
                       "this
        log.error
                               is
                                    а
                                         ErrorFilter
     \{\}",throwable.getCause\square.getMessage\square\square;
        ctx.set
     ☐ "error.status code", HttpServletResponse.SC INTERNA
     L SERVER ERROR□;
        ctx.set[]"error.exception",throwable.getCause[][];
        return null;
       }
   ThrowExceptionFilter
```

```
{
 "timestamp": 1481674993561,
 "status": 500,
 "error": "Internal Server Error",
 "exception": "java.lang.RuntimeException",
 "message": "Exist some errors..."
  }
 try {
   preRoute∏∏;
  } catch[ZuulException e[]{
   error[]e[];
   postRoute[];
   return;
  }
 try {
   route[];
```

```
} catch[]ZuulException e[]{
      error∏e∏;
      postRoute[];
      return;
   }
   try {
      postRoute∏;
   error∏e∏;
      return;
   }
  \square\square\square\square\square\square\square\square com.netflix.zuul.http.ZuulServlet\squareservice\square\square\square\square\square\square
____pre_route_post_____catch______catch
```



```
NOTION DE LA CONTRETA DEL CONTRETA DE LA CONTRETA DEL CONTRETA DE LA CONTRETA DEL CONTRETA DE LA CONTRETA DEL CONTRETA DE LA CONTRETA DEL CONTRETA DEL CONTRETA DE LA CONTR
Ondon One of the series of the
____ErrorFilter _____ error ______
SendErrorFilter
@Component
                            public class ErrorExtFilter extends SendErrorFilter {
                                       @Override
                                       public String filterType∏{
                                                return "error":
                                       @Override
                                       public int filterOrder∏{
                                                return 30; //∏∏ErrorFilter∏∏
                                       }
                                       @Override
                                       public boolean shouldFilter∏∏{
                                               //TODO _____post____
                                                return true;
                                       }
```

| ErrorExtFilter[][][]shouldFilter[][][][][][][][][][][][][][][][][][][] |
|--|
| |
| 000000000000000000000000000000000000Zuul000000 |
| Decom.netflix.zuul.FilterProcessor |
| |
| ●getInstance□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| ●setProcessor□FilterProcessor processor□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| ●processZuulFilter□ZuulFilter filter□□□□□□□□□□□ filter□□ |
| |
| ■ getFiltersByType [] String filterType [] [] [] [] [] [] [] |
| filterType API API filterOrder filte |
| |
| ●runFilters String sType filterType |
| getFiltersByType String filterType One of the content of the cont |
| ng processZuulFilter zuulFilter filter ng |
| ●preRoute□□□□□runFilters□"pre"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| ●route□□□□□runFilters□"route"□□□□□□□route□□□□□□□ |
| ●postRoute□□□□□runFilters□"post"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| <pre>●error[][][][]runFilters[]"error"[][][][][]error[][][][][]</pre> |
| |
| |
| public class DidiFilterProcessor extends FilterProcessor |
| { |
| @Override |

```
public Object processZuulFilter [] ZuulFilter filter []
                    throws ZuulException {
                                         try {
                                                        return super.processZuulFilter[];
                                          } catch□ZuulException e□{
                                                        RequestContext
                                         ctx=RequestContext.getCurrentContext[];
                                                        ctx.set□"failed.filter",filter□;
                                                       throw e;
                                          }
                                   }
                     }
              processZuulFilter \textsup ZuulFilter \text{filter} \textsup \text
@Component
                    public class ErrorExtFilter extends SendErrorFilter {
                            @Override
                            public String filterType□□{
                                  return "error";
                            @Override
                            public int filterOrder∏{
                                  return 30; //∏ErrorFilter∏∏
                            }
```

```
@Override
   public boolean shouldFilter□□{
    RequestContext
  ctx=RequestContext.getCurrentContext□□;
    ZuulFilter failedFilter= [] ZuulFilter []
                        ctx.get
  □"failed.filter"□;
    //____post____
    if ☐ failedFilter !=null && failedFilter.filterType
  \square.equals\square"post"\square\square{
     return true:
    }
    return false:
   }
  }
 □□□FilterProcessor.setProcessor□new DidiFilterProcessor□□□□□
SendErrorFilter
____SendErrorFilter_____
DDDDDSpring Cloud Zuul
```

```
\square\square\square\square\square\square\square\square/error\square\square\square\square\square\square\square\square
                                                                                   Spring
       /error
                              П
                                     П
                                           П
                                                  П
                                                         П
                                                                      П
                                                                                                     Boot
org.springframework.boot.autoconfigure.web.BasicErrorContr
oller
           @RequestMapping
           @ResponseBody
                            ResponseEntity<Map<String,Object>>
           public
                                                                                                            error
   □HttpServletRequest request□{
               Map<String,Object>
                                                                      body=getErrorAttributes
       ∏request,
                       isIncludeStackTrace request, Media Type. ALL r,;
               HttpStatus status=getStatus□reguest□;
                                          ResponseEntity<Map<String,Object>>
               return new
       ∏body,status∏;
           }
       org.spring framework.boot.autoconfigure.web. Error Attributes\\
Double Description
D
org.springframework.boot.autoconfigure.web.DefaultErrorAtt
ributes______Error_____Error______
@ConditionalOnMissingBean DefaultErrorAttributes DefaultErrorAttributes
\Pi\Pi
```

```
@ConditionalOnClass
                                                {
                                  П
 Servlet.class, Dispatcher Servlet.class }
    @ConditionalOnWebApplication
    @AutoConfigureBefore
 □WebMvcAutoConfiguration.class
    @Configuration
    public class ErrorMvcAutoConfiguration {
      @Autowired
      private ServerProperties properties;
      @Bean
      @ConditionalOnMissingBean
   □value=ErrorAttributes.class,search=SearchStrategy.CUR
   RENT
      public DefaultErrorAttributes errorAttributes □□{
        return new DefaultErrorAttributes□□;
      }
   DefaultErrorAttributes
public
              class
                       DidiErrorAttributes
                                           extends
 DefaultErrorAttributes {
      @Override
      public Map<String,Object> getErrorAttributes□
         RequestAttributes
                            requestAttributes,boolean
      includeStackTrace[[{
```

```
Map<String,Object>
   result=super.getErrorAttributes
   □requestAttributes,includeStackTrace□;
      result.remove[]"exception"[];
      return result;
     }
    }
@Bean
   public DefaultErrorAttributes errorAttributes □□{
     return new DidiErrorAttributes∏;
    }
  - ADDANDANA DA ANTA MARANDANA DA ANTA
- ANNONDO DE TOROS DE LA COMENZA DE LA C
zuul.<SimpleClassName>.<filterType>.disable=true
```

| zuul.AccessFilter.pre.disable=true |
|---|
| Zuul |
| |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| |
| 00000000API0000000000000000000000000000 |
| |
| application.properties application.yaml |

```
Zuul
config-server
_____config-server______
  route∏
  \bullet \square pom.xml \square \square \square zuul \square eureka \square config \square \square \square \square \square \square \square
   <parent>
     <groupId>org.springframework.boot</groupId>
     <artifactId>spring-boot-starter-parent</artifactId>
     <version>1.3.7.RELEASE
     <relativePath/>
   </parent>
   <dependencies>
     <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-zuul</artifactId>
     </dependency>
     <dependency>
      <groupId>org.springframework.cloud</groupId>
      <artifactId>spring-cloud-starter-
   eureka</artifactId>
     </dependency>
```

```
<groupId>org.springframework.cloud
       <artifactId>spring-cloud-starter-
    config</artifactId>
      </dependency>
    </dependencies>
    <dependencyManagement>
      <dependencies>
       <dependency>
         <groupId>org.springframework.cloud</groupId>
         <artifactId>spring-cloud-
      dependencies</artifactId>
         <version>Brixton.SR5
         <type>pom</type>
         <scope>import</scope>
       </dependency>
      </dependencies>
    </dependencyManagement>
   spring.application.name=api-gateway
    server.port=5556
    spring.cloud.config.uri=http://localhost:7001/
    eureka.client.serviceUrl.defaultZone=http://localhost:1
 111/eureka/
```

<dependency>

```
@EnableZuulProxy
    @SpringCloudApplication
    public class Application {
     public static void main [] String[] args [] {
                          SpringApplicationBuilder
       new
    \squareApplication.class\square.web\squaretrue\square.run\squareargs\square;
     }
     @Bean
     @RefreshScope
     @ConfigurationProperties□"zuul"□
     public ZuulProperties zuulProperties □ [
       return new ZuulProperties∏;
     }
    }
  gateway.properties
    zuul.routes.service-a.path=/service-a/**
    zuul.routes.service-a.serviceld=hello-service
    zuul.routes.service-b.path=/service-b/**
    zuul.routes.service-b.url=http://localhost:8001/
  bootstrap.properties | spring.application.name | | | | | | | | | | |
ndefaultnnnnnnnnnnnnnnnnnnnnnnnnn
```

```
______config-server_eureka-server_api-
service [][[][][] API [][] api-gateway-dynamic-route
c.c.c.ConfigServicePropertySourceLocator : Fetching
  config from server at:
    http://localhost:7001/
     c.c.c.ConfigServicePropertySourceLocator : Located
  environment: name=api-gateway,
    profiles=
 [default], label=master, version=1dab6126ca2972c5409fcb
 089934b057cf2bf77d
     b.c.PropertySourceBootstrapConfiguration: Located
  property source:
    CompositePropertySource[name='configService',prope
 rtySources=[MapPropertySource
    [name='overrides'], MapPropertySource
    [name='http://git.oschina.net/didispace/SpringCloud-
 Learning/spring cloud in action
    /config-repo/api-gateway.properties']]]
     com.didispace.Application
                                    : No active
  profile set, falling back to
    default profiles: default
  api-gateway-dynamic-route
{
```

```
"/service-a/**": "hello-service",
                                                 "/service-b/**": "http://localhost:8001/"
                    nnn path nnn/aaa/**nnn service-b nnnnn url nnnserviceld
zuul.routes.service-a.path=/aaa/**
                              zuul.routes.service-a.serviceld=hello-service
                             zuul.routes.service-b.path=/service-b/**
                              zuul.routes.service-b.url=hello-service
                    dynamic-route \( \) / refresh \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) \( \) 
"zuul.routes.service-b.serviceld",
                                                 "zuul.routes.service-b.url",
                                                 "zuul.routes.service-a.path"
                    On the service of the
serviceId חחחחחחחחחחחח service-bחחחחחחח
                    nnnnnnnnnnnnnnnnnnnnn API nnnnn/routesnnnn
{
                                                 "/aaa/**": "hello-service",
                                                 "/service-b/**": "hello-service"
```

} Described by the service of the serv service∏∏ JVM____Groovy_ ●□□□□□□Spring Boot□□□□□□api-gateway-dynamic-filter□ \bullet \square pom.xml \square \square \square zuul \square eureka \square groovy \square \square \square \square \square \square \square <parent> <groupId>org.springframework.boot</groupId> <artifactId>spring-boot-starter-parent</artifactId> <version>1.3.7.RELEASE <relativePath/> </parent> <dependencies>

```
<dependency>
   <groupId>org.springframework.cloud</groupId>
   <artifactId>spring-cloud-starter-zuul</artifactId>
 </dependency>
 <dependency>
   <groupId>org.springframework.cloud</groupId>
   <artifactId>spring-cloud-starter-
eureka</artifactId>
 </dependency>
 <dependency>
   <groupId>org.codehaus.groovy</groupId>
   <artifactId>groovy-all</artifactId>
 </dependency>
</dependencies>
<dependencyManagement>
 <dependencies>
   <dependency>
     <groupId>org.springframework.cloud
     <artifactId>spring-cloud-
 dependencies</artifactId>
     <version>Brixton.SR5</version>
     <type>pom</type>
     <scope>import</scope>
   </dependency>
 </dependencies>
</dependencyManagement>
```

```
nnnnnnnnnnnnn helloservicennnnnnnnnn
   spring.application.name=api-gateway
   server.port=5555
   eureka.client.serviceUrl.defaultZone=http://localhost:1
 111/eureka/
   zuul.routes.hello.path=/hello-service/**
   zuul.routes.hello.serviceld=hello-service
  ullet
application.properties□□□□□
   zuul.filter.root=filter
   zuul.filter.interval=5
  DDzuul.filter.root
@ConfigurationProperties[]"zuul.filter"[]
   public class FilterConfiguration {
     private String root;
     private Integer interval;
     public String getRoot∏∏{
      return root:
     }
     public void setRoot∏String root∏{
      this.root=root:
     }
```

```
public Integer getInterval□□{
         return interval;
       }
       public void setInterval□Integer interval□{
         this.interval=interval:
       }
     }
   @EnableZuulProxy
     @EnableConfigurationProperties
 ∏{FilterConfiguration.class}∏
     @SpringCloudApplication
     public class Application {
         public static void main ☐ String [] args ☐ {
                                   SpringApplicationBuilder
           new
       \squareApplication.class\square.web\squaretrue\square.run\squareargs\square;
         }
         @Bean
         public FilterLoader filterLoader | FilterConfiguration
     filterConfiguration [ {
           FilterLoader filterLoader=FilterLoader.getInstance
       \Pi\Pi;
           filterLoader.setCompiler 

new GroovyCompiler
       try {
```

```
GroovyFileFilter□□□;
        FilterFileManager.init□
           filter Configuration.get Interval \verb| | | |,
           filterConfiguration.getRoot□□+"/pre",
           filterConfiguration.getRoot∏+"/post"∏;
      } catch□Exception e□{
        throw new RuntimeException[]e[];
      }
      return filterLoader:
     }
   }
  ____5__ API ____ filter/pre _filter/post_____Groovy_
חחחחחחחחחחחחחחחחחחחחחחחחחחnnnnnzuul.filter.interval
_____eureka-server_hello-service_
☐ API ☐☐☐☐☐☐☐☐☐☐http://localhost:5555/hello-service/hello☐☐☐☐
● []filter/pre [] [] [] [] [] [] [] PreFilter.groovy []
  class PreFilter extends ZuulFilter {
```

FilterFileManager.setFilenameFilter

new

```
@Override
                                     String filterType∏∏{
                                              return "pre"
                                     }
                                     @Override
                                     int filterOrder∏∏{
                                              return 1000
                                     }
                                     @Override
                                     boolean shouldFilter∏∏{
                                              return true
                                     }
                                     @Override
                                     Object run∏∏{
                                              HttpServletRequest
                           request=RequestContext.getCurrentContext
                           □□.getRequest□□
                                              log.info∏"this is a pre filter: Send {} request to {}",
                                                                 request.getMethod∏,
                                                                request.getRequestURL[].toString[][]
                                              return null
                                     }
                   ONDO DE LA CONTRETA DEL CONTRETA DE LA CONTRETA DEL CONTRETA DE LA CONTRETA DEL CONTRETA DE LA CONTRETA DEL CONTRETA DE LA CONTRETA DEL CONTRETA DE LA CONTRETA DEL CONTRETA DEL CONTRETA DEL CONTRETA DE LA CONTRETA DEL CONTRETA DE LA CONTRETA DEL CON
DDD PreFilter.groovy
```

Logger log=LoggerFactory.getLogger PreFilter.class

```
com.didispace.filter.pre.PreFilter : this is a pre
   filter: Send GET request to
    http://localhost:5555/ddd/hello
   class PostFilter extends ZuulFilter{
                         log=LoggerFactory.getLogger
      Logger
   □PostFilter.class□
      @Override
      String filterType□□{
       return "post"
      }
      @Override
      int filterOrder∏∏{
        return 2000
      }
      @Override
      boolean shouldFilter∏∏{
        return true
      }
      @Override
      Object run□□{
       log.info["this is a post filter: Receive response"[]
       HttpServletResponse
    response=RequestContext.getCurrentContext
    \squaregetResponse\square
```

```
response.getOutputStream \ \ \square \ \ .print \ \ \square \ \ ",I
             zhaiyongchao"[]
                       response.flushBuffer∏∏
                  }
              }
         ADDODO DO TORO DE LA COLLEGA D
zhaiyongchao∏
         org.springframework.cloud.netflix.zuul.ZuulFilterInitializer □□
false
```

_8 □ □□□□□□□Spring Cloud Config



```
OSpring Cloud Config
   ●□□□□□□Spring Boot□□□□□□config-server□□□pom.xml□□
<parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.3.7.RELEASE
        <relativePath/> <!--lookup parent from repository-
     ->
     </parent>
     <dependencies>
        <dependency>
          <groupId>org.springframework.cloud
          <artifactId>spring-cloud-config-
      server</artifactId>
        </dependency>
     </dependencies>
     <dependencyManagement>
        <dependencies>
          <dependency>
             <groupId>org.springframework.cloud
         pld>
             <artifactId>spring-cloud-
         dependencies</artifactId>
```

<version>Brixton.SR5

```
<type>pom</type>
            <scope>import</scope>
         </dependency>
        </dependencies>
    </dependencyManagement>
   Spring Cloud Config
    @EnableConfigServer
    @SpringBootApplication
    public class Application {
      public static void main [] args [] args [] {
                             SpringApplicationBuilder
        new
    □Application.class□.web□true□.run□args□;
   spring.application.name=config-server
    server.port=7001
    spring.cloud.config.server.git.uri=http://git.oschina.net/
 didispace/SpringCloud-Learning/
    spring.cloud.config.server.git.searchPaths=spring clou
 d in action/config-repo
    spring.cloud.config.server.git.username=username
    spring.cloud.config.server.git.password=password
   ●spring.cloud.config.server.git.uri□□□Git□□□□□
```

| spring.cloud.config.server.git.searchPaths |
|--|
| |
| <pre>espring.cloud.config.server.git.username[][]Git[][][][]</pre> |
| ●spring.cloud.config.server.git.password□□□Git□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| |
| |
| config-server Git |
| http://git.oschina.net/didispace/SpringCloud- |
| Learning/spring_cloud_in_action/ |
| |
| •didispace.properties |
| •didispace-dev.properties |
| •didispace-test.properties |
| •didispace-prod.properties |
| 004000000000from000000000000000000000000000 |
| ●from=git-default-1.0 |
| ●from=git-dev-1.0 |
| ●from=git-test-1.0 |
| •from=git-prod-1.0 |
| Gitmasterfrom1.0 |
| config-label-test conf |
| |
| |
| \{application}/{profile}[/{label}] |

```
●/{application}-{profile}.yml
               ●/{label}/{application}-{profile}.yml
               •/{application}-{profile}.properties

/{label}/{application}-{profile}.properties
               Understand the second control of the se
 Online config-label-test online didispace on prodontino
url:http://localhost:7001/didispace/prod/configlabel-test
{
                                     "name": "didispace",
                                     "profiles":[
                                            "prod"
                                     "label": "config-label-test",
                                     "version":
                       "4c4f3909b0499b8518aba1f76e8a90b0dbad535d",
                                     "propertySources":[
                                             {
                                                           "name":
                                            "http://git.oschina.net/didispace/SpringCloud-
                                            Learning/spring cloud in action/config-
                                            repo/didispace-prod.properties",
                                                           "source": {
                                                                  "from": "git-prod-2.0"
                                                           }
                                             },
```

```
"name":
    "http://git.oschina.net/didispace/SpringCloud-
Learning/spring_cloud_in_action/config-
repo/didispace.properties",
    "source": {
        "from": "git-default-2.0"
        }
    }
    ]
    ]
    ]
    ]
    []
    ]
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    []
    [
```

1 parent 8f4e9938bc

s.c.a.AnnotationConfigApplicationContext: Refreshing org.springframework.context.annotation.AnnotationConfigApplicationContext@7d09c9:

startup date[Fri Sep 16 21:56:43 CST 2016]; root of context hierarchy

| o.s.c.c.s.e.NativeEnvironmentRepository : Adding |
|--|
| property source: |
| file:/C:/Users/ADMINI [] 1/AppData/Local/Temp/config |
| repo-914347082723810766/spring_cl |
| oud_in_action/config-repo/didispace-prod.properties |
| o.s.c.c.s.e.NativeEnvironmentRepository : Adding |
| property source: |
| file:/C:/Users/ADMINI [] 1/AppData/Local/Temp/config |
| repo-914347082723810766/spring_cl |
| oud_in_action/config-repo/didispace.properties |
| s.c.a.AnnotationConfigApplicationContext : Closing |
| org.spring framework.context.annotation. Annotation Co |
| nfigApplicationContext@7d09c9: |
| startup date[Fri Sep 16 21:56:43 CST 2016]; root o |
| context hierarchy |
| config-server Git Git Git |
| |
| $http://localhost: 7001/didispace/prod/config-label-test \verb $ |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| not pull remote for config-label-test |
| config-server |
| .c.s.e.MultipleJGitEnvironmentRepository : Could no |
| pull remote for config-label-test |
| <pre>□current ref=Ref[refs/heads/config-label-test=</pre> |
| 4c4f3909b0499b8518aba1f76e8a90b0dbad535d] , re |
| mote: |
| http://git.oschina.net/didispace/SpringCloud-Learning |

s.c.a.AnnotationConfigApplicationContext: Refreshing org.springframework.context.annotation.AnnotationCo nfigApplicationContext@55c747ad: startup date[Mon Sep 19 10:51:04 CST 2016]; root of context hierarchy o.s.c.c.s.e.NativeEnvironmentRepository : Adding property source: file:/C:/Users/ADMINI | 1/AppData/Local/Temp/configrepo-7514536618307405051/spring c loud in action/config-repo/didispace-prod.properties o.s.c.c.s.e.NativeEnvironmentRepository Adding property source: file:/C:/Users/ADMINI | 1/AppData/Local/Temp/configrepo-7514536618307405051/spring c loud in action/config-repo/didispace.properties s.c.a.AnnotationConfigApplicationContext: Closing org.spring framework.context.annotation. Annotation ConfigApplicationContext@55c747ad: startup date[Mon Sep 19 10:51:04 CST 2016]; root of context hierarchy ●□□□□Spring Boot□□□□□□config-client□□□pom.xml□□□□□□

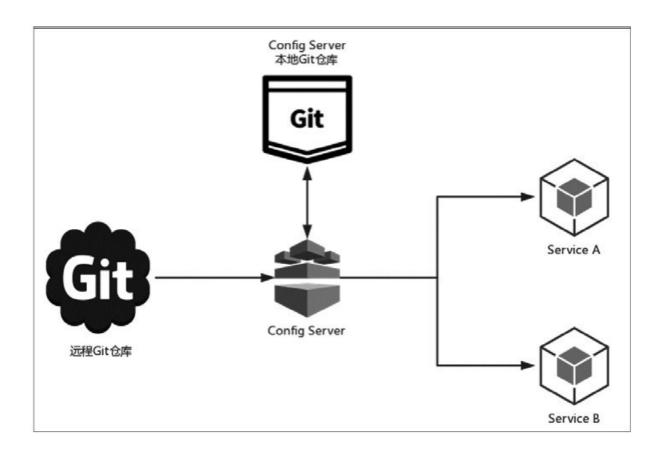
```
<parent>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-parent</artifactId>
   <version>1.3.7.RELEASE
   <relativePath/> <!--lookup parent from repository-
->
</parent>
<dependencies>
   <dependency>
     <groupId>org.springframework.boot</groupId>
     <artifactId>spring-boot-starter-web</artifactId>
   </dependency>
   <dependency>
     <groupId>org.springframework.cloud</groupId>
     <artifactId>spring-cloud-starter-
 config</artifactId>
   </dependency>
</dependencies>
<dependencyManagement>
   <dependencies>
     <dependency>
        <groupId>org.springframework.cloud
     pld>
        <artifactId>spring-cloud-
     dependencies</artifactId>
        <version>Brixton.SR5
        <type>pom</type>
```

```
<scope>import</scope>
         </dependency>
       </dependencies>
    </dependencyManagement>
   ●□□Spring Boot□□□□□□□□□□
    @SpringBootApplication
    public class Application {
      public static void main

☐String[]args
☐{
                            SpringApplicationBuilder
       new
    \squareApplication.class\square.web\squaretrue\square.run\squareargs\square;
      }
    }
   ● □ □ bootstrap.properties □ □ □ □ □ □ □ □ □ config-server □
spring.application.name=didispace
    spring.cloud.config.profile=dev
    spring.cloud.config.label=master
    spring.cloud.config.uri=http://localhost:7001/
    server.port=7002
   ●spring.application.name□□□□□□□□[application]□□□
   ●spring.cloud.config.profile□□□□□□□□□[profile]□□□
   ●spring.cloud.config.uri□□□□□config-server□□□□
   server
____jar_____jar_______jar_______jar_______
```

```
bootstrap.properties config-server config-server
@RefreshScope
    @RestController
    public class TestController {
     @Value[]"${from}"[
     private String from;
     @RequestMapping[]"/from"[]
     public String from □ [ {
      return this.from:
     }
    }
  @RefreshScope
    @RestController
    public class TestController {
     @Autowired
     private Environment env;
     @RequestMapping[]"/from"[]
     public String from∏∏{
      return env.getProperty[]"from","undefined"[];
     }
    }
```

| config-client http://localhost:7002/from |
|--|
| |
| |
| |
| Spring Cloud Config |
| ●□□ Git □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| ●Config Server□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| ● |
| ●Service A□Service B□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |



- 2.Config Server
 - 3. git clone config Server config Server
- 4.Config Server Spring Application Context Grant Git Grant Git Grant Gra

Config Server

<u>Git</u>∏∏∏

| Spring Cloud Config |
|--|
| |
| Hook |
| Online Git |
| |
| DSpring Cloud ConfigDDDGitDDDGitDDDDGitDDDDDDDDD |
| Config Server application.properties appli |
| spring.cloud.config.server.git.uri |
| |
| spring.cloud.config.server.git.uri=http://git.oschina.net/ |
| didispace/SpringCloud-Learning/ |
| spring.cloud.config.server.git.username=username |
| spring.cloud.config.server.git.password=password |
| |
| file:///0000000000000000000000000000000000 |
| |
| spring.cloud.config.server.git.uri=file://\${user.home}/c |
| onfig-repo |
| \${user.home} |
| 00000000000000000000000000000000000000 |
| |
| |
| {application}[[{profile}[[{label}[[][[][[][[][[][[][[][[][[][[][[][[][[] |
| Config Server Git URI |
| {application} |

| spring.cloud.config.server.git.uri=http://git.oschina.net/ |
|--|
| didispace/{application} |
| spring.cloud.config.server.git.username=username |
| spring.cloud.config.server.git.password=password |
| {application}Config Server |
| Config Server spring.application.name |
| {application} |
| {label}Git"/"{label} |
| HTTP_URL"""URIURIURI |
| 0000 Git 0000000000000000000000000000000 |
| URI |
| ● □ □ □ □ □ □ □ □ □ Git □ □ □ □ □ □ □ □ □ □ □ □ |
| http://git.oschina.net/didispace/member-service[] |
| ●□□□□ □□□□□□-config□□□□Git□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| http://git.oschina.net/didispace/member-serviceconfig[] |
| |
| spring.cloud.config.server.git.uri=http://git.oschina.net/didisp |
| ace/{application}-config |
| |
| Config Server |
| |
| <pre>[][][]{application}/{profile}[][][][][]</pre> |
| spring.cloud.config.server.git.uri=http://git.oschina.net/ |
| didispace/config-repo |
| spring.cloud.config.server.git.repos.dev.pattern=dev/* |
| spring.cloud.config.server.git.repos.dev.uri=file://home |
| /git/config-repo |

spring.cloud.config.server.git.repos.test=http://git.osch ina.net/test/config-repo spring.cloud.config.server.git.repos.prod.pattern=prod/ pp*,online/oo* spring.cloud.config.server.git.repos.prod.uri=http://git. oschina.net/prod/confi g-repo ____ spring.cloud.config.server.git.uri _____ _____dev_test_prod____dev ____ dev/*____dev_modev_____application חחחחחחחחחחחחחחחחConfig Server חחחחחחחחח Gitnnnn spring.cloud.config.server.git.uri http://git.oschina.net/didispace/SpringCloud-Learning/ [] []

| spring.cloud.config.server.git.uri=http://git.oschina.net, |
|---|
| didispace/SpringCloud-Learning/ |
| spring.cloud.config.server.git.searchPaths= |
| {application} |
| |
| $http://git.oschina.net/didispace/SpringCloud-Learning/ \verb $ |
| |
| |
| Config Server |
| username[]password[][][][][][][][][][][][][][][][][][][] |
| spring.cloud.config.server.git.uri=http://git.oschina.net/ |
| didispace/application |
| spring.cloud.config.server.git.username=username |
| spring.cloud.config.server.git.password=password |
| HTTPSSHKeyGit |
| |
| <u>SVN</u> |
| Config ServerDDDGitDDDDDDDDSVNDDDDDDDDDDD |
| Config Server□□□□Git□□□□□□□□SVN□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| <pre><dependency></dependency></pre> |
| <pre><groupid>org.tmatesoft.svnkit</groupid></pre> |
| <artifactid>svnkit</artifactid> |
| <pre><version>1.8.10</version></pre> |
| |
| \/uepenuency/ |

| ● application.properties SVN SV |
|---|
| |
| spring.cloud.config.server.svn.uri=svn://localhost:443/ |
| didispace/config-repo |
| spring.cloud.config.server.svn.username=username |
| spring.cloud.config.server.svn.password=password |
| Config ServerSVN |
| |
| |
| |
| Git_SVNConfig Server |
| |
| |
| |
| spring.cloud.config.server.git.basedir |
| spring.cloud.config.server.svn.basedir |
| |
| |
| Spring Cloud Config |
| |
| spring.profiles.active=native,Config Server [|
| src/main/resource |
| spring.cloud.config.server.native.searchLocations |
| |
| Spring Cloud Config |
| |

| 00000000 URI 000000000000000000000000000 |
|--|
| spring.cloud.config.server.git.uri=http://git.oschina.net/didisp |
| ace/{application}-config |
| o.s.c.c.s.e.JGitEnvironmentRepository : Could not |
| fetch remote for master remote: |
| http://git.oschina.net/didispace/app-config |
| <pre>[[] [] [] [] [] [] Spring Cloud Config [] [] [] spring-</pre> |
| bootactuator /health |
| org.springframework.cloud.config.server.config.ConfigServer |
| $Health Indicator \verb $ |
| |
| http://git.oschina.net/didispace/app-config [][[][[][[][[][[][[][[][[][[][[][[][[][|
| /health |
| "configServer": { |
| "status": "DOWN", |
| "repository": { |
| "application": "app", |
| "profiles": "default" |
| }, |
| "error": |
| "org.springframework.cloud.config.server.environment. |
| NoSuchLabelException:No such label: master" |
| } |
| |
| $http://git.oschina.net/didispace/app-config \verb $ |

| DOWN |
|--|
| |
| Gitapp-config |
| |
| check-repo-config |
| spring.cloud.config.server.git.uri=http://git.oschina.net/ |
| didispace/{applicatio n}-config |
| spring.cloud.config.server.git.username=username |
| spring.cloud.config.server.git.username=password |
| spring.cloud.config.server.health.repositories.check.na |
| me=check-repo |
| spring.cloud.config.server.health.repositories.check.lab |
| el=master |
| spring.cloud.config.server.health.repositories.check.pro |
| files=default |
| repositoriesMap |
| |
| ●name[][][] |
| ●label□□□□□ |
| ●profiles□□□□ |
| |
| spring.cloud.config.server.health.enabled=false |
| |
| Config Server |
| Despring.cloud.config.server.overrides |

| Map |
|---|
| spring.cloud.config.server.overrides.name=didi |
| spring.cloud.config.server.overrides.from=shanghai |
| Spring CloudSpring Cloud |
| Config Server |
| Cloud |
| Config Server |
| |
| |
| |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| <pre> <pre> <pre> </pre> <pre> <pre> <pre> </pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> </pre> <pre> </pre> <pre> </pre> <pre> <pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre> |
| <pre><groupid>org.springframework.boot</groupid></pre> |
| <artifactid>spring-boot-starter-</artifactid> |
| security |
| |
| |
| |
| INFO 22028[|
| main]b.a.s.AuthenticationManagerConfiguration: Using |
| default security password: 1a32a848-da0c-4590-9c58- |
| e860be8c50dd |
| |

security.user.name=user security.user.password=37cc5635-559b-4e6f-b633-7e932b813f73 _____config-server____________ spring.cloud.config.username=user spring.cloud.config.password=37cc5635-559b-4e6fb633-7e932b813f73 On the configuration of the control spring.datasource.username=didi spring.datasource.password= {cipher}dba6505baa81d78bd08799d8d4429de499bd4c20 53c0 5f029e7cfbf143695f5b Spring Cloud Config

```
On Spring Cloud Config
Java Cryptography Extension
README.txt
                        local policy.jar
                        US export policy.jar
                 Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

Description:

D
●/keynnnnnnn
                 ●/encrypt□□□□□body□□□□□□□□
                 ●/decrypt□□□□□body□□□□□□□□
                 On the status of the status of
                         {
                         }
                                         "description": "No key was installed for encryption
                        service",
                                         "status": "NO KEY"
                 encrypt.key=didispace
```

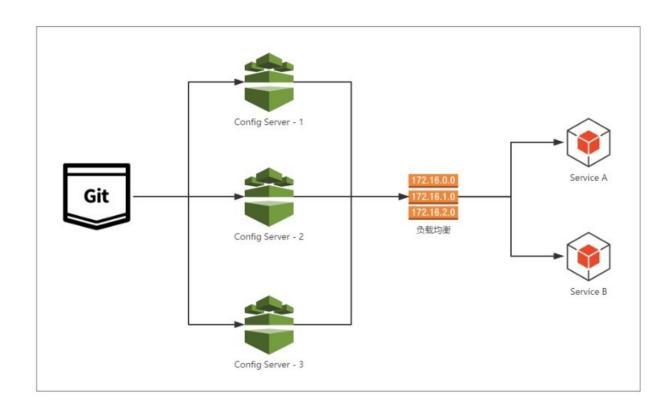
```
{
     "status": "OK"
   }
  \square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square\square/encrypt\\decrypt
$ curl localhost:7001/encrypt-d didispace
     3c70a809bfa24ab88bcb5e1df51cb9e4dd4b8fec8830
  1eb7a18177f1769c849ae9c9f29400c9204
   80be2c99406ae28c7
     $ curl localhost:7001/decrypt-d
   3c70a809bfa24ab88bcb5e1df51cb9e4dd4b8fec88301
 eb7a18177f1769c849ae9c9f29400c920480be
   2c99406ae28c7
     didispace
  Spring Cloud Config
RSANDONDONONO
nnnnnnnnnnnnjDK 1.4nnnnnnnnnnnnnnnnn
□%JAVA HOME%\bin\keytool.exe□
```

| \$ keytool-genkeypair-alias config-server-keyalg RSA |
|--|
| keystore config-server.keystore |
| |
| |
| |
| [Unknown]: zhaiyongchao |
| |
| [Unknown]: company |
| |
| [Unknown]: organization |
| |
| [Unknown]: city |
| |
| [Unknown]: province |
| |
| [Unknown]: china |
| CN=zhaiyongchao,OU=company,O=organization,L=cit |
| y,ST=province,C=china[][][] |
| [□]: y |
| <config-server> </config-server> |
| |
| |
| 00000000000000000000000000000000000000 |
| -storepass-keypass-long-long-long-long-long-long-long-long |
| |

\$ keytool-genkeypair-alias config-server-keyalg RSA \

```
-dname
  "CN=zhaiyongchao,OU=company,O=organization,L=city
  ,ST=province,C=china" \
     -keypass 222222 \
     -keystore config-server.keystore \
     -storepass 111111 \
  validity
    $ keytool-genkeypair-alias config-server-keyalg RSA \
     -dname
  "CN=zhaiyongchao,OU=company,O=organization,L=city
  ,ST=province,C=china" \
     -keypass 222222 \
     -keystore config-server.keystore \
     -storepass 111111 \
     -validity 365 \
  configserver.keystore
encrypt.key-store.location=file://${user.home}/config-
 server.keystore
    encrypt.key-store.alias=config-server
    encrypt.key-store.password=111111
    encrypt.key-store.secret=222222
  □□□□□config-server.keystore□□□□□□□src/main/resource□□
```

| server.keystore |
|----------------------------|
| |
| ENCRYPT_KEY_STORE_LOCATION |
| ENCRYPT_KEY_STORE_ALIAS |
| ENCRYPT_KEY_STORE_PASSWORD |
| ENCRYPT_KEY_STORE_SECRET |
| |
| |





Cloud Config
 Cloud Config
 Cloud Config



| Spring Cloud Config |
|---|
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Spring Cloud Config |
| DDDDDDbootstrap.properties |
| spring.cloud.config.uriSpring Cloud |
| Config DDDDDDDDhttp://localhost:8888 |
| |
| $spring.application.name \verb spring.cloud.config.profile \verb $ |
| |
| spring.application.name=didispace |
| spring.cloud.config.profile=dev |
| spring.cloud.config.uri=http://localhost:7001/ |
| |
| |
| |
| |
| |
| |
| |
| |
| |
| Spring Cloud Config Server |
| Spring Cloud Config Server |
| Spring Cloud Config Server |
| Spring Cloud Config Server |
| Spring Cloud Config Server Config-client Config-server Config-server |

```
<dependency>
         <groupId>org.springframework.cloud</groupId>
         <artifactId>spring-cloud-config-server</artifactId>
       </dependency>
       <dependency>
         <groupId>org.springframework.cloud</groupId>
         <artifactId>spring-cloud-starter-
     eureka</artifactId>
       </dependency>
     </dependencies>
               application.properties
                                                      П
eureka.client.serviceUrl.defaultZone
     spring.application.name=config-server
     server.port=7001
     # 0000000
     eureka.client.serviceUrl.defaultZone=http://localhost:1
 111/eureka/
     # Git∏∏∏
     spring.cloud.config.server.git.uri=http://git.oschina.net/
 didispace/SpringCloud-Learning/
     spring.cloud.config.server.git.searchPaths=spring clou
 d in action/config-repo
     spring.cloud.config.server.git.username=username
     spring.cloud.config.server.git.password=password
   ●□□□□□□□□□@EnableDiscoveryClient□□□□□□config-server
```

```
@EnableDiscoveryClient
       @EnableConfigServer
       @SpringBootApplication
       public class Application {
         public static void main

☐String[]args
☐{
                                            SpringApplicationBuilder
            new
       □Application.class□.web□true□.run□args□;}
    ●□□□□□□□□http://localhost:1111/□□□□Eureka Server□□□
\square\square\square\square\squareconfig-server\square\square\square\square\square\square\square
```

| Instances currently registered with Eureka | | | |
|--|---------|--------------------|--|
| Application | AMIs | Availability Zones | Status |
| CONFIG-SERVER | n/a (1) | (1) | UP (1) - PC-201602152056:config-server:7001 |

● □ config-client □ pom.xml □ □ □ spring-cloud-starter-eureka □ _____config-server_____

<dependencies> <dependency> <groupId>org.springframework.boot</groupId> <artifactId>spring-boot-starter-web</artifactId> </dependency> <dependency> <groupId>org.springframework.cloud

```
<artifactId>spring-cloud-starter-
                       config</artifactId>
                            </dependency>
                            <dependency>
                                  <groupId>org.springframework.cloud</groupId>
                                  <artifactId>spring-cloud-starter-
                       eureka</artifactId>
                            </dependency>
                 </dependencies>
                  . . . . . .
           ● [bootstrap.properties [] [] [] [] [] []
                 spring.application.name=didispace
                 server.port=7002
                 eureka.client.serviceUrl.defaultZone=http://localhost:1
      111/eureka/
                 spring.cloud.config.discovery.enabled=true
                 spring.cloud.config.discovery.serviceId=config-server
                 spring.cloud.config.profile=dev
           spring.cloud.config.discovery.serviceId [[] [] Config Server [[]
Description: Description of the configuration of th
● [ ] [ ] [ ] [ ] @EnableDiscoveryClient [ ] [ ] [ ] [ ] config-
server
                 @EnableDiscoveryClient
```

```
@SpringBootApplication
     public class Application {
      public static void main

∏String[]args

{
                               SpringApplicationBuilder
        new
     □Application.class□.web□true□.run□args□;
     }
   ● ______Controller _____Git ______
     @RefreshScope
     @RestController
     public class TestController {
      @Value□"${from}"□
      private String from;
      @RequestMapping[]"/from"[]
      public String from∏∏{
        return this.from;
      }
   ullet
http://localhost:1111/0000Eureka Server
```

| | 0.000 | | |
|---------------|---------|--------------------|--|
| Application | AMIs | Availability Zones | Status |
| CONFIG-SERVER | n/a (1) | (1) | UP (1) - PC-201602152056:config-server:7001 |
| DIDISPACE | n/a (1) | (1) | UP (1) - PC-201602152056:didispace:7002 |

| | | _http://localho | st:7002/fro | m[][][][| Git |
|--|--|--|---------------------------------|------------|------------------|
| □□□didisp | ace-dev.p | roperties[[[[[| from | ∥"git-de | v-1.0"□ |
| | | | 700 | | |
| | | | | | |
| Sprin | g Cloud Co | onfig | 100000000 | 000000 |][Config |
| Server[][[| | | Co | nfig Serv | /er[[[[[[|
| | | | | □□Config | Server[] |
| | 00000000 | 10000000000 | | | □□Config |
| Server 🛮 🗎 | | 0000000000 |]bootstrap. | propertie | es 🛮 🖺 🖺 🗎 |
| spring.clc | oud.config. | failFast=true[| | | |
| | |]Cor | ıfig Server∐ | | |
| | | | | □Control | ler[[[[[|
| org | .springfrar | mework.beans | s.factory.Bea | anCreation | onExcep |
| | | | | | |
| tion: | Error | creating | bean | with | name |
| | | creating
tController': | | | name
utowired |
| 'scoped | | tController': | | | |
| 'scoped
depend | lTarget.tes
lencies fail | tController': | Injection | | |
| 'scoped
depend | lTarget.tes
lencies fail | tController':
led; | Injection
=true | | utowired |
| 'scoped
depend
□□spr | lTarget.tes
lencies fail
ring.cloud.
llllllllllll | tController':
led; | Injection
=true | of a | utowired |
| 'scoped
depend
spr | Target.tes
lencies fail
ing.cloud.
100000000 | tController':
led;
config.failFast | Injection
=true | of a | utowired |
| 'scoped
depend
□□spr
□□□□□□□□
java | ITarget.tes
lencies fail
ring.cloud.
1000000000000000000000000000000000000 | tController':
led;
config.failFast
000000000000 | Injection =true Con tion: Cou | of a | utowired |
| 'scoped depend | ITarget.tes Iencies fail Ing.cloud. IDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD | tController':
led;
config.failFast
000000000000000000000000000000000000 | Injection =true Con tion: Cou | of a | utowired |
| 'scoped depend | ITarget.tes Iencies fail Ing.cloud. Ing.cloud. Ing.cloud. Ing.llleg ySource a | tController':
led;
config.failFast
000000000000000000000000000000000000 | Injection =true | of a | utowired |
| 'scoped depend depend Spr | Target.tes lencies fail ing.cloud. looooooooooooooooooooooooooooooooooo | tController': led; config.failFast config.failFast config.failFast config.failFast config.failFast config.failFast config.failfast config.failfast | Injection =true | of a | utowired |
| 'scoped depend depend Spring S | ITarget.tes Iencies fail Ing.cloud. Ing.cloud. Ing.cloud. Ing.llleg Ing.llleg Ing.llnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn | tController': led; config.failFast | Injection =true | of a | utowired |

```
● 🛮 🔻 🗎 🔻 pom.xml 🔻 🖂 spring-retry 🗘 spring-boot-starter-
<dependencies>
        <dependency>
         <groupId>org.springframework.retry</groupId>
         <artifactId>spring-retry</artifactId>
        </dependency>
        <dependency>
         <groupId>org.springframework.boot</groupId>
         <artifactId>spring-boot-starter-aop</artifactId>
        </dependency>
    </dependencies>
   Server_______6___6_____6______
c.c.c.ConfigServicePropertySourceLocator : Fetching
   config from server at:
    http://PC-201602152056:7001/
      c.c.c.ConfigServicePropertySourceLocator : Fetching
   config from server at:
    http://PC-201602152056:7001/
      c.c.c.ConfigServicePropertySourceLocator : Fetching
   config from server at:
    http://PC-201602152056:7001/
```

| c.c.c.ConfigServicePropertySourceLocator : Fetching |
|---|
| config from server at: |
| http://PC-201602152056:7001/ |
| c.c.c.ConfigServicePropertySourceLocator : Fetching |
| config from server at: |
| http://PC-201602152056:7001/ |
| c.c.c.ConfigServicePropertySourceLocator : Fetching |
| config from server at: |
| http://PC-201602152056:7001/ |
| o.s.boot.SpringApplication : Application |
| startup failed |
| java.lang.lllegalStateException: Could not locate |
| PropertySource and the fail fast property is set,failing |
| |
| •spring.cloud.config.retry.multiplier |
| |
| • spring.cloud.config.retry.initial-interval |
| 1.1_{\Box |
| •spring.cloud.config.retry.max-interval |
| |
| •spring.cloud.config.retry.max-attempts |
| |
| |
| |

| lem:lem:lem:lem:lem:lem:lem:lem:lem:lem: |
|---|
| |
| ●□□□Config Server□□GET□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| ■ {label}master |
| □/{application}-{profile}.yml |
| □/{application}-{profile}.properties |
| ■ [{label} |
| <pre>□/{label}/{application}-{profile}.yml</pre> |
| <pre>□/{application}/{profile}[/{label}]</pre> |
| <pre>□/{label}/{application}-{profile}.properties</pre> |
| ullet |
| ■spring.application.name |
| ■spring.cloud.config.profile□□□□□□[{profile}□□ |
| ■spring.cloud.config.label |
| |
| Spring Cloud Config |
| |
| |
| ●config-repo□□□□Git□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| from |
| ●config-server□□□□Git□□□□□□ |
| ■ config-client [] [] config-server [] [] [] [] [] [] [] [] [] [|
| didispace |
| config-repo/didispace-dev.properties [] from [] [] [|

| $\verb $ |
|--|
| $dev-1.0 \verb $ |
| repo/didispace-dev.properties $\ \ \ \ \ \ \ \ \ \ \ \ \ $ |
| $\verb from=git-dev-2.0 $ |
| □□□git-dev-1.0□ |
| config-client |
| config-client_pom.xmlspring-boot-starter-actuator |
| /refresh |
| <dependency></dependency> |
| <pre><groupid>org.springframework.boot</groupid></pre> |
| <artifactid>spring-boot-starter-actuator</artifactid> |
| |
| ●□□□□config-client□□□□□http://localhost:7002/from□□□□□ |
| |
| ■ □ □ Git □ □ config-repo/didispace-dev.properties □ □ □ □ from □ |
| |
| ●□□□□□http://localhost:7002/from□□□□□□□□□□□□ |
| ●□□ POST □□□□□http://localhost:7002/refresh□□□□□□□□□ |
| from |
| [|
| "from" |
|] |
| ●□□□□□http://localhost:7002/from□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| GitGitWeb Hook Git |
| /refresh |
| |

| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD | us[[[[[[[[[[[| | |
|--|---------------|--|--|
| | | | |

□9□ □□□□Spring Cloud Bus

_____Spring Cloud Bus___



ullet

| ullet |
|---|
| ●□□Web□□□□□□ |
| |
| ullet |
| |
| ●ActiveMQ |
| ●Kafka |
| ●RabbitMQ |
| ●RocketMQ |
| ● |
| |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| |
| RabbitMQ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| RabbitMQAMQP |
| RabbitMQ |
| |
| AMQP Amale |
| |
| |
| |
| |
| |
| |

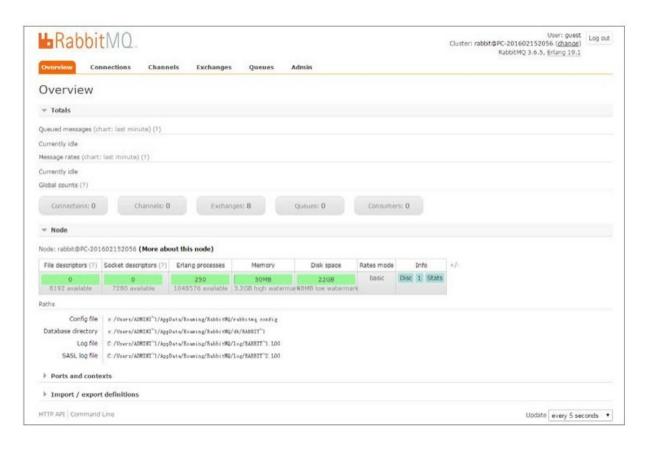
| ●Connection□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
|---|
| ●Channel□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| $Channel \verb $ |
| |
| ●Producer□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| ●Consumer□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| 1Channel_ |
| 2Exchange |
| 3Queue |
| 4Routing KeyExchange_Queue |
| 5Exchange |
| 6.Exchange |
| |
| Exchange |
| 1.Direct |
| Key_abc |
| 2.TopicKey* |
| □□□□□□□abc.#□□abc.def.ghi,abc.*□□□abc.def□ |
| 3.FanoutKey |
| |
| RabbitMQ |
| |
| 1.Exchange |
| $2.Queue \square \square$ |
| 3delivery_mode=> 2_1 |

| <pre>[][Exchange]Quet</pre> | re[][][| | | ∥Bind | ling[[| | |
|-----------------------------|---------------------|------------|---------|-----------|------------|---------------|-------|
| Exchange Queue Queue | | | | | | | |
| | | | | | | | |
| | Ш | | | | | | |
| ☐ RabbitMQ | | | | | | | |
| https://www.rabbitmq. | com/do | wnloa | ad.htm | ا | | | |
| | | | | | | | |
| | abbitM | Q Ser | ver[[[| | | | |
| ●Erlang/OTP 19.1 | | | | | | | |
| ●RabbitMQ Server | 3.6.5 | | | | | | |
| □Windows □□□□ | | | | | | | |
| 1.00 Erlang |]htt | p://wv | ww.erl | ang.o | rg/dow | ınloa | ads 🛮 |
| | | | | | | | |
| 2. 🛮 🗎 Rab | bitMQ | | | | | | |
| https://www.rabbitmq. | com/do | wnloa | ad.htm | าl⊟⊟ex | œ□□□□ | | |
| 3.00000000000000 |][] | | | | | | |
| 4.RabbitMQ Serve | -—
r | | | | | | |
| | | | | | | | |
| 名称 描述 Multi-protocol | open source messagi | ing broker | | 状态
已启动 | 启动类型
自动 | 登录为
本地系统 | |
| | | | | | | | |
| Windows | | | | | | | |
| | | | | | | | |
| ☐ RABBITMQ_BASE ☐☐[| | |]E:\ser | ver\ra | abbitm | ı q □□ | |
| □□RabbitMQ□□□ | | | | | | | |
| □Mac OS X□□□ | | | | | | | |
| ☐Mac OS X□□□brev | N | | □□Rab | bitMQ | | | |
| | | | | | | | |

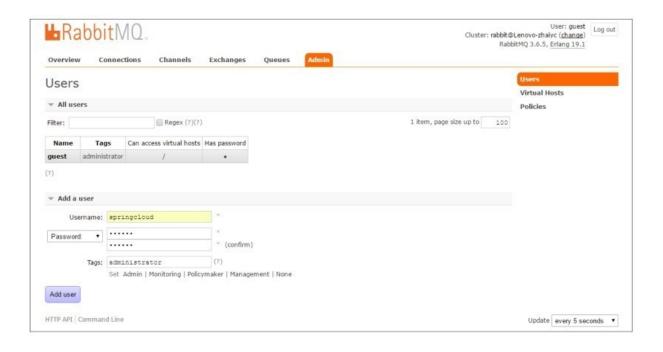
```
1. brew \square \square \square brew update \square
            2.□□Erlang□□□brew install erlang□□□
            3. □ RabbitMQ Server □ □ brew install rabbitm □ □
            □□□□□□□□□RabbitMQ Server□□□□□□/usr/local/sbin□□□□□□□
_____bash profile_profile___
                  PATH=$PATH:/usr/local/sbin
            \square\square\square\square\square\square\square\squarerabbitmq-server\square\square\square\square\squareRabbitMQ\square\square\square\square\square\square
            ∏Ubuntu∏∏∏
            1.□□Erlang□□□apt-get install erlang□□□
            2. DO DO DO DE LA COMPTION DEL COMPTION DE LA COMPTION DEL COMPTION DE LA COMPTION DEL COMPTION DE LA COMPTION DEL COMPTION DE LA COMPTION DEL COMPTION DE LA COMPTION DEL COMPTION DE LA 
                                                  'debhttp://www.rabbitmq.com/debian/testing
                  echo
      main' l
                        sudo tee /etc/apt/sources.list.d/rabbitmq.list
            3. ☐ APT☐ ☐ package list☐ ☐ sudo apt-get update☐ ☐ ☐
            4. □ Rabbit Server □ □ sudo apt-get install rabbitmg-server
Rabbit∏∏
            ■□□ rabbitmq-plugins enable rabbitmq_management □□□
nn Webnnnnnnnnnnnnnnnnnn
                  > rabbitmq-plugins enable rabbitmq_management
                  The following plugins have been enabled:
                        mochiweb
                        webmachine
                        rabbitmq web dispatch
```

amqp_client
rabbitmq_management_agent
rabbitmq management

Applying plugin configuration to rabbit@PC-201602152056...started 6 plugins.



Connections Channels Exchanges Queues Channels RabbitMQ Server



□□□Tags□□□RabbitMQ□□□□□□□□□□□□

- ●none□□□□□management plugin□
- ■□□□□□□□□AMQP□□□virtual hosts□
- **■**□□□□□virtual hosts□□queues□exchanges □ bindings□
- **■**□□□□□□□channels □ connections□
- ■□□□□□□□virtual hosts□"□□"□□□□□□□□□□□□□□□□virtual hosts

- ■□□□□□□□□□□□virtual hosts□□□policies□parameters□
- **■**□□□□virtual hosts□□□□□□□□□□virtual hosts□
- **■**□□□□□□connections□channels□
- **■**_______clustering_memory____
- ■□□□□□□□□□virtual hosts□□□□□□□□
- ●administrator:policymaker monitoring monitoring

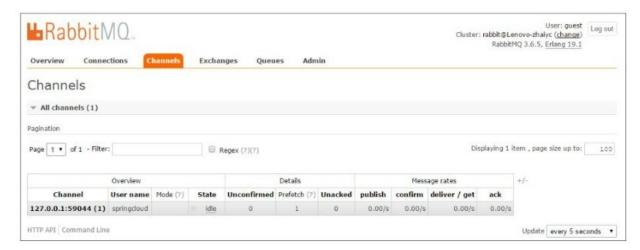
| ■□□□□□virtual hosts□ |
|--|
| ■ □□□□□□users□ |
| ■ □ □ □ □ □ □ □ permissions □ |
| ■ □ □ □ □ □ □ connections □ |
| |
| |
| Spring BootRabbitMQ |
| |
| Spring Boot RabbitMQ |
| Starter POMs Output Description Descripti |
| |
| ●□□□Spring Boot□□□□□rabbitmq-hello□ |
| ● pom.xml □□□□□□□□□□ spring-boot-starter-amqp □□□ |
|]RabbitMQ∏ |
| <pre><parent></parent></pre> |
| <pre><groupid>org.springframework.boot</groupid></pre> |
| <artifactid>spring-boot-starter-parent</artifactid> |
| <pre><version>1.3.7.RELEASE</version></pre> |
| <relativepath></relativepath> lookup parent from repository-</td |
| -> |
| |
| <dependencies></dependencies> |
| <dependency></dependency> |
| <pre><groupid>org.springframework.boot</groupid></pre> |
| <artifactid>spring-boot-starter-</artifactid> |
| amqp |
| |

```
</dependency>
      <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
      </dependency>
    </dependencies>
  DODO Springcloud
spring.application.name=rabbitmq-hello
    spring.rabbitmq.host=localhost
    spring.rabbitmq.port=5672
    spring.rabbitmq.username=springcloud
    spring.rabbitmq.password=123456
  AmqpTemplate
@Component
    public class Sender {
     @Autowired
     private AmqpTemplate rabbitTemplate;
     public void send∏∏{
      String context="hello "+new Date□□;
      System.out.println∏"Sender : "+context∏;
      this.rabbitTemplate.convertAndSend
    □"hello",context□;
```

```
}
  @Component
   @RabbitListener gueues = "hello"
   public class Receiver {
     @RabbitHandler
     public void process String hello {
      System.out.println□"Receiver: "+hello□;
     }
   }
  @Configuration
   public class RabbitConfig {
     @Bean
     public Queue helloQueue∏∏{
      return new Queue ["hello" [];
     }
  @SpringBootApplication
   public class HelloApplication {
     public static void main [] String[] args [] {
      SpringApplication.run HelloApplication.class,args;
```

```
}
  ullet
    @RunWith[Spring]Unit4ClassRunner.class[]
    @SpringApplicationConfiguration
 □classes=HelloApplication.class
    public class HelloApplicationTests {
     @Autowired
     private Sender sender;
     @Test
     public void hello∏throws Exception {
       sender.send∏∏;
     }
    }
  חחחחחחחחחחחחחחחחחחRabbitMQ Server
springcloud□□□□
    o.s.a.r.c.CachingConnectionFactory
                                Created
                                        new
 connection: SimpleConnection@29836d32
    [delegate=amqp://springcloud@127.0.0.1:5672/]
```





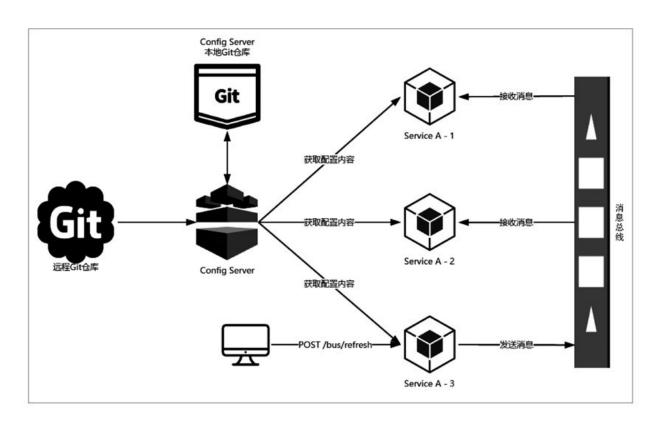
Sender: hello Sun Sep 25 11:06:11 CST 2016

Receiver: hello Sun Sep 25 11:06:11 CST 2016

| RabbitMQ Server |
|--|
| |
| |
| |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Spring Cloud Config |
| DDDDDDDDDSpring Cloud ConfigDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| Web Hook |
| |
| |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| <u>□□Spring Cloud Bus</u> |
| <pre></pre> |
| |
| |
| |
| <pre></pre> |
| Spring Cloud Spring Boot Spring Spring Boot RabbitMQ Spring Cloud Bus RabbitMQ Cloud Bus Cloud B |
| □ Spring Cloud □ Spring Boot □ □ □ □ □ □ Spring Boot □ RabbitMQ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ |
| □□Spring Cloud □□ Spring Boot□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| □Spring Cloud □Spring Boot □□□□□□□□□□Spring Boot □RabbitMQ□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| □Spring Cloud □ Spring Boot □□□□□□□□ Spring Boot □RabbitMQ□□□□□Spring Cloud Bus□□RabbitMQ□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |

| config-repo/didispace-dev.properties |
|--|
| ●□□config-client-eureka□□□ |
| ■ [] pom.xml [] [] spring-cloud-starter-bus-amqp [] [] [] |
| springboot-starter-actuator |
| <dependency></dependency> |
| <pre><groupid>org.springframework.cloud</groupid></pre> |
| <artifactid>spring-cloud-starter-bus-</artifactid> |
| amqp |
| |
| ullet |
| spring.rabbitmq.host=localhost |
| spring.rabbitmq.port=5672 |
| spring.rabbitmq.username=springcloud |
| spring.rabbitmq.password=123456 |
| ■□□config-server-eureka□□□□□□config-client-eureka□□□□ |
| $\verb $ |
| /bus/refresh |
| o.s.b.a.e.mvc.EndpointHandlerMapping : Mapped |
| "{[/bus/refresh],methods=[POST]}" onto public void |
| org.springframework.cloud.bus.endpoint.RefreshBusEn |
| dpoint.refresh[]java.lang.String[] |
| ● □ □ □ □ □ config-client-eureka □ /from □ □ □ □ □ □ □ |
| configrepo/didispace-dev.properties [[] from [] [] |
| ●□□□□□config-repo/didispace-dev.properties□□from□□□□□ |
| POST /bus/refresh |
| ●□□□□□□□□□□□□config-client-eureka□/from□□□□□□□□□□ |
| <pre>□□□□□□ config-repo/didispace-dev.properties □□from□□□</pre> |

On the state of th

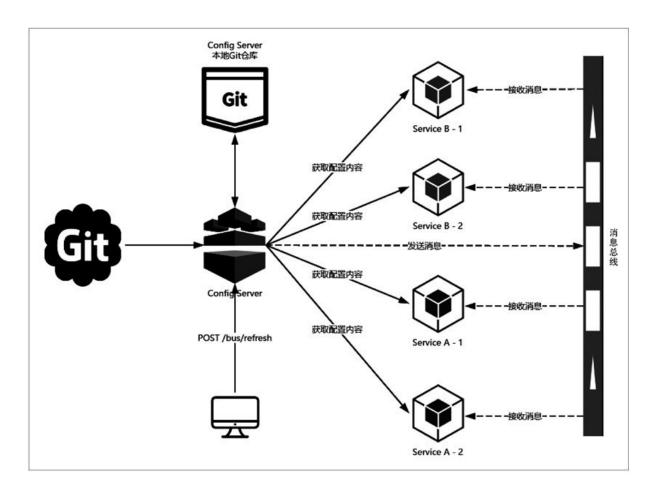


Config Server

Grand Serv

Ondone "Service A" Ondone Git Ondone Ondone

| DDDDDConfig Server |
|--|
| |
| DOITED DOING THE POST DOING THE DOIN |
| Web Hook |
| Web Hook |
| |
| Spring Cloud Bus_/bus/refresh |
| |
| |
| Spring Cloud Bus [][][][][][][][] bus/refresh [][][][] |
| destination [][[][[][][][][][][][][][][][][][][][] |
| $destination = customers: 9000 \verb $ |
| |
| Spring Cloud Netflix_Eureka |
| |
| \${spring.cloud.client.hostname}:\${spring.application.name} |
| :\${spring.application.instance id:\${server.port}}} |
| |
| destination [][[][[][[][[][[][[][[][[][[][[][[][[][|
| |
| destination=customers:** destination=customers:** destination=customers:** |
| |
| |
| □□Spring Cloud Bus□/bus/refresh□□□□□□□□□□□□□□□□□□ |
| |
| |



- 1. Config Server Cloud Bus Cloud Bus
- 2./bus/refresh

RabbitMQ□□

| Sprin | g Cloud | Bus 🛮 🖺 R | labbitMC | 0 0 0 0 | 🛮 🖟 Spi | ing | Воо | t 🛮 |
|------------|-----------|-----------|----------|---------|---------|-----|-----|-----|
| Connection | nFactory | / 🗆 🗆 🗆 | Spring | Cloud | Bus [] | | | |
| spring.ral | obit.mq[] | □□□Spring | Boot 🗆 🗆 | 000000 | | | | |
| | | | | | | | | |

| 属性名 | 说明 | 默认值 |
|---|------------------------------|-----------|
| spring.rabbitmq.addresses | 客户端连接的地址,有多个的时候使用逗 | |
| | 号分隔,该地址可以是 IP 与 Port 的结合 | |
| spring.rabbitmq.cache.channel.checkout-timeout | 当缓存已满时, 获取 Channel 的等待时间, | |
| | 单位为毫秒 | |
| spring.rabbitmq.cache.channel.size | 缓存中保持的 Channel 数量 | |
| spring.rabbitmq.cache.connection.mode | 连接缓存的模式 | CHANNEL |
| spring.rabbitmq.cache.connection.size | 缓存的连接数 | |
| spring.rabbitmq.connection-timeout | 连接超时参数,单位为毫秒;设置为"0" | |
| | 代表无穷大 | |
| spring.rabbitmq.dynamic | 默认创建一个 AmqpAdmin 的 Bean | true |
| spring.rabbitmq.host | RabbitMQ 的主机地址 | localhost |
| spring.rabbitmq.listener.acknowledge-mode | 容器的 acknowledge 模式 | |
| spring.rabbitmq.listener.auto-startup | 启动时自动启动容器 | true |
| spring.rabbitmq.listener.concurrency | 消费者的最小数量 | |
| spring.rabbitmq.listener.default-requeue-rejected | 投递失败时是否重新排队 | true |
| spring.rabbitmq.listener.max-concurrency | 消费者的最大数量 | |
| spring.rabbitmq.listener.prefetch | 在单个请求中处理的消息个数, 它应该大 | |
| | 于等于事务数量 | |
| spring.rabbitmq.listener.retry.enabled | 不论是不是重试的发布 | false |
| spring.rabbitmq.listener.retry.initial-interval | 第一次与第二次投递尝试的时间间隔 | 1000 |
| spring.rabbitmq.listener.retry.max-attempts | 尝试投递消息的最大数量 | 3 |
| spring.rabbitmq.listener.retry.max-interval | 两次尝试的最大时间间隔 | 10000 |
| spring.rabbitmq.listener.retry.multiplier | 上一次尝试时间间隔的乘数 | 1.0 |
| spring.rabbitmq.listener.retry.stateless | 不论重试是有状态的还是无状态的 | true |
| spring.rabbitmq.listener.transaction-size | 在一个事务中处理的消息数量。为了获得 | |
| | 最佳效果,该值应设置为小于等于每个请 | |
| | 求中处理的消息个数,即 spring. | |
| | rabbitmq.listener.prefetch的值 | |
| spring.rabbitmq.password | 登录到 RabbitMQ 的密码 | |
| spring.rabbitmq.port | RabbitMQ 的端口号 | 5672 |
| spring.rabbitmq.publisher-confirms | 开启 Publisher Confirm 机制 | false |
| spring.rabbitmq.publisher-returns | 开启 Publisher Return 机制 | false |
| spring.rabbitmq.requested-heartbeat | 请求心跳超时时间,单位为秒 | |
| spring.rabbitmq.ssl.enabled | 启用 SSL 支持 | false |

| 属性名 | 说明 | 默认值 |
|---|-------------------------------|-------|
| spring.rabbitmq.ssl.key-store | 保存 SSL 证书的地址 | |
| spring.rabbitmq.ssl.key-store-password | 访问 SSL 证书的地址使用的密码 | |
| spring.rabbitmq.ssl.trust-store | SSL 的可信地址 | |
| spring.rabbitmq.ssl.trust-store-password | 访问 SSL 的可信地址的密码 | |
| spring.rabbitmq.ssl.algorithm | SSL 算法,默认使用 Rabbit 的客户端算法库 | |
| spring.rabbitmq.template.mandatory | 启用强制消息 | false |
| spring.rabbitmq.template.receive-timeout | receive()方法的超时时间 | 0 |
| spring.rabbitmq.template.reply-timeout | sendAndReceive()方法的超时时间 | 5000 |
| spring.rabbitmq.template.retry.enabled | 设置为 true 的时候 RabbitTemplate 能 | false |
| | 够实现重试 | |
| spring.rabbitmq.template.retry.initial-interval | 第一次与第二次发布消息的时间间隔 | 1000 |
| spring.rabbitmq.template.retry.max-attempts | 尝试发布消息的最大数量 | 3 |
| spring.rabbitmq.template.retry.max-interval | 尝试发布消息的最大时间间隔 | 10000 |
| spring.rabbitmq.template.retry.multiplier | 上一次尝试时间间隔的乘数 | 1.0 |
| spring.rabbitmq.username | 登录到 RabbitMQ 的用户名 | Ê |
| spring.rabbitmq.virtual-host | 连接到 RabbitMQ 的虚拟主机 | |

Kafka

Kafka∏∏

| Kafka |
|---------------------------------|
| ApacheKafkaScalaLinkedIn |
| |
| Kafkannnnn-nnnnnnnnnnnnnnnnnnnn |

| ●□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
|---|
| |
| ullet |
| ullet |
| ●Java_PHP_Python |
| ullet |
| |
| Kafka |
| ●Broker:Kafka□□□□□□□□□□□□□□□□□□□□□□Broker□ |
| ●Topic□□□□RabbitMQ□Queue□□□□□□□□□Kafka□□□□□□□ |
| Topic |
| Broker |
| ● Partition: Partition ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ |
| DDDDD PartitionDDD Partition DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| ●Producer□□□□□□□□□□□□□□Kafka Broker□ |
| ●Consumer□□□□□□Kafka Broker□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| ● Consumer Group [] Consumer [] [] [] [] [] [] [] [] [] [|
| Consumer |
| |
| |
| |
| KafkaKafkaKafka |
| Kafka 🗆 🗆 🗆 🗆 🗆 |
| |
| |
| http://kafka.apache.org/downloads.html [] [] [] [] [] [Kafka- |
| |

```
0.10.0.1
  kafka
    +-bin
     +-windows
    +-config
    +-libs
    +-logs
    +-site-docs
  □□Kafka□□□□□□□ZooKeeper□□□□□□bin□config□□□□□□□
Kafka [][[][[][] ZooKeeper [][[][][] bin [][[][] Kafka []
■ □ ZooKeeper □ □ □ □ zookeeper-server-start
config/zookeeper.properties
[2016-09-28
                          Reading
             08:05:34,8491INFO
  configuration from: config\zookeeper.
   properties
 □org.apache.zookeeper.server.quorum.QuorumPeerConfig□
    [2016-09-28
                     08:05:34,850]INFO
  autopurge.snapRetainCount set to 3□org.apache.
```

| zookeeper.serv | er.DatadirCleanupMa | anager⊔ |
|----------------------|-----------------------|---------------------|
| [2016-09-28 | | 08:05:34,851]INFO |
| autopurge.purge | nterval set to 0∏org | .apache. |
| zookeeper.serv | er.DatadirCleanupMa | anager[] |
| [2016-09-28 | 08:05:34,851]INFO | Purge task is not |
| scheduled.∐org.a | pache.zookeeper. | |
| server.Datadir0 | leanupManager[] | |
| [2016-09-28 | 08:05:34,852]WARN | Either no config or |
| no quorum define | ed in config, | |
| running | in standa | alone mode |
| □org.apache.zooke | eper.server.quorum. | QuorumPeerMain[] |
| [2016-09-28 | 08:05:34,868 | JINFO Reading |
| configuration from | m: config\zookeeper. | |
| properties | | |
| □org.apache.zooke | eper.server.quorum. | QuorumPeerConfig[] |
| [2016-09-28 | 08:05:34,869]INFO |) Starting server |
| □org.apache.zook | keeper.server. | |
| ZooKeeperServ | erMain[] | |
| ••• | | |
| | 08:05:34,940]INFC | binding to port |
| 0.0.0.0/0.0.0.0:21 | | |
| zookeeper.serv | er.NIOServerCnxnFa | ctory□ |
| | | ooKeeper 🛮 🗎 🗎 |
| config/zookeeper.pro | perties | |
| | | |
| config/zookeeper.pro | perties client | Port |
| ZooKeeper□ | | |

| ● 🛮 🖺 Kafka 🖟 🖺 🖟 kafka-server-start |
|--|
| $config/server.properties \verb $ |
| |
| $\verb zookeeper.connect ZooKeeper $ |
| $\cdots ZooKeeper \cdots \cdo$ |
| ZooKeeper |
| zookeeper.connect=127.0.0.1:3000,127.0.0.1:3001,127.0.0. |
| $1:3002 \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ $ |
| |
| ● 🛮 🖶 Topic 🖺 🖺 🖺 kafka-topicscreatezookeeper |
| localhost:2181replication-factor 1partitions 1topic test□□ |
| |
| |
| Topic |
| kafka-topics |
| |
| • 🛮 🗎 🗎 🗎 🖺 🗎 kafka-console-producerbroker-list |
| localhost:9092topic test kafka-console-producer |
| Kafka |
| |
| |
| • [] [] [] [] kafka-console-consumerzookeeper |
| localhost:2181topic testfrom-beginning [] kafka- |
| consoleconsumer |
| |
| Kafka |

□□Spring Cloud Bus

| Kafka spring-cloud-starter-bus-amqp |
|--|
| RabbitMQ RabbitMQ Kafka Kafka RabbitMQ Rabb |
| $cloud\text{-}starter\text{-}bus\text{-}amqp \verb \verb spring\text{-}cloud\text{-}starter\text{-}bus\text{-}kafka \verb $ |
| pom.xml dependency |
| <dependency></dependency> |
| <pre><groupid>org.springframework.cloud</groupid></pre> |
| <artifactid>spring-cloud-starter-bus-</artifactid> |
| kafka |
| |
| 00000 Kafka 00000000000000000000000000000000000 |
| Rabbit MQ [Kafka] [] [] [] [] [] [] Zoo Keeper [Kafka] [] [] [] [] [] [] [] [] [] [] [] [] [] |
| $\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $ |
| client |
| config-server |

```
2016-09-28 22:11:29.627 INFO 15144 --- [
 o.s.c.s.b.k.KafkaMessageChannelBinder : Using kafka topic for outbound:
 springCloudBus
     2016-09-28 22:11:29.642 INFO 15144 --- [-localhost:2181]
 org.IOItec.zkclient.ZkEventThread : Starting ZkClient event thread.
     016-09-28 22:11:30.290 INFO 15144 --- [
 o.s.i.kafka.support.ProducerFactoryBean : Using producer properties =>
 {bootstrap.servers=localhost:9092, linger.ms=0, acks=1, compression.type=none,
 batch.size=16384}
     2016-09-28 22:11:30.298 INFO 15144 --- [
 o.a.k.clients.producer.ProducerConfig : ProducerConfig values:
     2016-09-28 22:11:30.322 INFO 15144 --- [
                                               main]
 o.s.c.s.b.k.KafkaMessageChannelBinder$1 : Adding
 {message-handler:outbound.springCloudBus} as a subscriber to the 'springCloudBusOutput'
 channel
     2016-09-28 22:11:30.322 INFO 15144 --- [
                                               mainl
 o.s.integration.channel.DirectChannel : Channel
 'config-server:7001.springCloudBusOutput' has 1 subscriber(s).
     2016-09-28 22:11:30.322 INFO 15144 --- [
 o.s.c.s.b.k.KafkaMessageChannelBinder$1 : started outbound.springCloudBus
     2016-09-28 22:11:31.465 INFO 15144 --- [
                                               mainl
 s.i.k.i.KafkaMessageDrivenChannelAdapter : started
 org.springframework.integration.kafka.inbound.KafkaMessageDrivenChannelAdapter@4178
     2016-09-28 22:11:31.467 INFO 15144 --- [
                                               mainl
 o.s.c.s.b.k.KafkaMessageChannelBinder$7 : Adding
 {message-handler:inbound.springCloudBus.anonymous.8b9e6c7b-6a50-48c5-b981-8282a0d5a
 30b} as a subscriber to the 'bridge.springCloudBus' channel
     2016-09-28 22:11:31.467 INFO 15144 --- [
 o.s.c.s.b.k.KafkaMessageChannelBinder$7 : started
 inbound.springCloudBus.anonymous.8b9e6c7b-6a50-48c5-b981-8282a0d5a30b
    _____Kafka_____
springCloudBus∏Topic∏
    □□□□□□□springCloudBus□Topic□
```

□□□□□□□□spring-cloud-starter-bus-kafka□□□config-client□

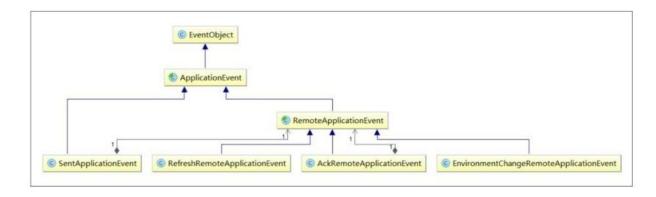
```
2016-09-28 22:43:55.067 INFO 6136 --- [
o.s.c.s.b.k.KafkaMessageChannelBinder : Using kafka topic for outbound:
springCloudBus
    2016-09-28 22:43:55.078 INFO 6136 --- [-localhost:2181]
org.IOItec.zkclient.ZkEventThread : Starting ZkClient event thread.
   2016-09-28 22:50:38.584 INFO 828 --- [
                                                mainl
o.s.i.kafka.support.ProducerFactoryBean : Using producer properties =>
{bootstrap.servers=localhost:9092, linger.ms=0, acks=1, compression.type=none,
batch.size=16384}
    2016-09-28 22:50:38.592 INFO 828 --- [
o.a.k.clients.producer.ProducerConfig : ProducerConfig values:
    2016-09-28 22:50:38.615 INFO 828 --- [
                                                mainl
o.s.c.s.b.k.KafkaMessageChannelBinder$1 : Adding
{message-handler:outbound.springCloudBus} as a subscriber to the 'springCloudBusOutput'
channel
    2016-09-28 22:50:38.616 INFO 828 --- [
o.s.integration.channel.DirectChannel : Channel
'didispace: 7002.springCloudBusOutput' has 1 subscriber(s).
    2016-09-28 22:50:38.616 INFO 828 --- [
o.s.c.s.b.k.KafkaMessageChannelBinder$1 : started outbound.springCloudBus
    2016-09-28 22:50:39.162 INFO 828 --- [
                                                main]
s.i.k.i.KafkaMessageDrivenChannelAdapter : started
org.springframework.integration.kafka.inbound.KafkaMessageDrivenChannelAdapter@60cf
    2016-09-28 22:50:39.162 INFO 828 --- [
                                                mainl
o.s.c.s.b.k.KafkaMessageChannelBinder$7 : Adding
{message-handler:inbound.springCloudBus.anonymous.f8fc9c0c-ccd3-46dd-9537-07198f4ee
216} as a subscriber to the 'bridge.springCloudBus' channel
    2016-09-28 22:50:39.163 INFO 828 --- [
                                                main]
o.s.c.s.b.k.KafkaMessageChannelBinder$7 : started
inbound.springCloudBus.anonymous.f8fc9c0c-ccd3-46dd-9537-07198f4ee216
```

_config-client_____ 2016-09-29 08:20:34.361 INFO 21256---[kafkabinder-1] o.s.cloud.bus.event.RefreshListener : Received remote refresh request. Keys refreshed[from] RefreshListener One of the second second control of the second seco **Kafka**∏∏ ____Kafka_ZooKeeper________ $\prod_{n=1}^{\infty} 10^n$ ___/bus/refresh_____Spring Cloud Bus [] [] [] springCloudBus [Topic []] [] [] [] kafkaconsole-consumer--zookeeper localhost:2181--topic $springCloudBus \square \square \square \square springCloudBus \square \square$

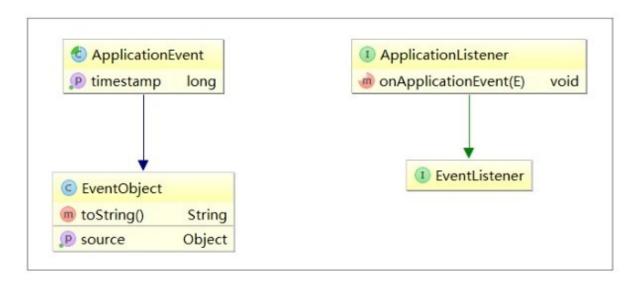
```
contentType "application/json"
     {
       "type": "RefreshRemoteApplicationEvent",
       "timestamp": 1475073160814,
       "originService": "config-server:7001",
       "destinationService": "*:**",
       "id": "bbfbf495-39d8-4ff9-93d6-174873ff7299"
     }
     contentType "application/json"
     {
       "type": "AckRemoteApplicationEvent",
       "timestamp": 1475073160821,
       "originService": "config-server:7001",
       "destinationService": "*:**",
       "id": "1f794774-10d6-4140-a80d-470983c6c0ff".
       "ackId": "bbfbf495-39d8-4ff9-93d6-174873ff7299",
       "ackDestinationService": "*:**",
       "event":
   "org.springframework.cloud.bus.event.RefreshRemoteAp
   plicationEvent"
     }
     contentType "application/json"
     {
       "type": "AckRemoteApplicationEvent",
       "timestamp": 1475075467554,
```

```
"originService": "didispace:7002",
    "destinationService": "*:**",
    "id": "7560151e-f60c-49cd-8167-b691e846ad08",
    "ackId": "21502725-28f5-4d19-a98a-f8114fa4f1dc",
    "ackDestinationService": "*:**",
    "event":
  "org.springframework.cloud.bus.event.RefreshRemoteAp
  plicationEvent"
   }
  ■ type □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
Refresh Remote Application Event \\
AckRemoteApplicationEvent
AckRemoteApplicationEvent
  nnnnnnnnnnnnnn RabbitMQ nnnnnnnnnnnnnnnnn
destinationService didispace:**
   contentType "application/json"
   {
     "type": "RefreshRemoteApplicationEvent",
     "timestamp": 1475131215007,
```

| "originService": "config-server:7001", |
|--|
| "destinationService": "didispace:**", |
| "id": "667fe948-e9b2-447f-be22-3c8acf647ead" |
| } |
| ●id□□□□□□□ |
| ☐ ☐ ☐ ☐ ☐ ☐ RefreshRemoteApplicationEvent ☐ |
| $Ack Remote Application Event \ \ \square \ \ \square$ |
| AckRemoteApplicationEvent |
| ● ackld:Ack 🛛 🖺 🗎 🗎 🖺 🖺 🖺 🖺 🖺 🖺 🖺 🖺 |
| AckRemoteApplicationEvent ackId |
| ${\sf RefreshRemoteApplicationEvent} \ \ \square \ \ id \ \ \square \ \ \square$ |
| RefreshRemoteApplicationEvent |
| ●ackDestinationService:Ack □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| *:**\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ |
| ● event:Ack |
| $RefreshRemoteApplicationEvent ~ \verb \verb \verb \verb \verb \verb \verb \verb$ |
| client |
| ackDestinationService $\ \ ^*:^{**} \ \ \ \ \ \ \ \ \ $ config-client $\ \ \ \ \ \ $ |
| RefreshRemoteApplicationEvent |
| |
| |
| |
| RefreshRemoteApplicationEvent |
| $Ack Remote Application Event \verb $ |
| Cloud Bus |



RefreshRemoteApplicationEvent AckRemoteApplicationEvent RemoteApplicationEvent RemoteApplic



| |] ApplicationEvent | |
|---|---|--|
| EventObject |]timestamp | |
| DDDDDDDDDDDDDDsource | :00000000000000000000000000000000000000 | |
| <pre>ApplicationEvent [</pre> |] [] RemoteApplicationE | vent [|
| RefreshRemoteApplicatio | nEvent | |
| | | |
| ●□□□□□ Spring □□□□ |] ApplicationListene | er |
| JDK □ EventListener □ □ | ☐ ☐ ApplicationListener ☐ | пппп |
| |]□onApplicationEvent□E even | |
| | tener ApplicationEver | |
| | | |
| | | |
| | | |
| | | |
| ApplicationEventPublisher | ApplicationEventMulticaster | |
| publishEvent(ApplicationEvent) void | addApplicationListener(ApplicationListener) | void |
| | addApplicationListener(ApplicationListener<? >) addApplicationListenerBean(String) removeApplicationListener(ApplicationListener<? >) | void
void |
| publishEvent(ApplicationEvent) void | addApplicationListener(ApplicationListener<? >) addApplicationListenerBean(String) | void |
| publishEvent(ApplicationEvent) void | m addApplicationListener(ApplicationListener) m addApplicationListenerBean(String) m removeApplicationListener(ApplicationListener) m removeApplicationListenerBean(String) | void
void
void |
| publishEvent(ApplicationEvent) void | m addApplicationListener(ApplicationListener) m addApplicationListenerBean(String) m removeApplicationListener(ApplicationListener) m removeApplicationListenerBean(String) m removeAllListeners() m multicastEvent(ApplicationEvent) | void
void
void
void
void |
| publishEvent(ApplicationEvent) void publishEvent(Object) void | m addApplicationListener(ApplicationListener) m addApplicationListenerBean(String) m removeApplicationListener(ApplicationListener) m removeApplicationListenerBean(String) m removeAllListeners() m multicastEvent(ApplicationEvent) | void
void
void
void
void
void |
| publishEvent(ApplicationEvent) void publishEvent(Object) void | m addApplicationListener(ApplicationListener) m addApplicationListenerBean(String) m removeApplicationListener(ApplicationListener) m removeApplicationListenerBean(String) m removeAllListeners() m multicastEvent(ApplicationEvent) m multicastEvent(ApplicationEvent, ResolvableType) | void
void
void
void
void
void |
| publishEvent(ApplicationEvent) void publishEvent(Object) void Spring ApplicationEventMulticas | addApplicationListener(ApplicationListener) addApplicationListenerBean(String) removeApplicationListener(ApplicationListener) removeApplicationListenerBean(String) removeAllListeners() multicastEvent(ApplicationEvent) multicastEvent(ApplicationEvent, ResolvableType) ApplicationEventPub | void void void void void void void void |
| publishEvent(ApplicationEvent) void void publishEvent(Object) Spring ApplicationEventMulticas: ApplicationEventPublishe | addApplicationListener(ApplicationListener) addApplicationListenerBean(String) removeApplicationListener(ApplicationListener) removeApplicationListenerBean(String) removeAllListeners() multicastEvent(ApplicationEvent) multicastEvent(ApplicationEvent, ResolvableType) ApplicationEventPub removeAllListeners() multicastEvent(ApplicationEvent, ResolvableType) | void void void void void void void void |
| publishEvent(ApplicationEvent) void void publishEvent(Object) Spring ApplicationEventMulticas: ApplicationEventPublishe ApplicationEvent event | m addApplicationListener(ApplicationListener) m addApplicationListenerBean(String) m removeApplicationListener(ApplicationListener) m removeApplicationListenerBean(String) m removeAllListeners() m multicastEvent(ApplicationEvent) m multicastEvent(ApplicationEvent, ResolvableType) ApplicationEventPub ter | lisher [] shEvent |
| publishEvent(ApplicationEvent) void publishEvent(Object) void Spring ApplicationEventMulticas ApplicationEventPublishe ApplicationEvent event ApplicationEventMulticas | addApplicationListener(ApplicationListener) addApplicationListenerBean(String) removeApplicationListener(ApplicationListener) removeApplicationListenerBean(String) removeApplicationListenerBean(String) removeAllListeners() multicastEvent(ApplicationEvent) multicastEvent(ApplicationEvent, ResolvableType) ApplicationEventPub re | void void void void void void void void |
| publishEvent(ApplicationEvent) void publishEvent(Object) void Spring ApplicationEventMulticas ApplicationEventPublishe ApplicationEvent event ApplicationEventMulticas | m addApplicationListener(ApplicationListener) m addApplicationListenerBean(String) m removeApplicationListener(ApplicationListener) m removeApplicationListenerBean(String) m removeAllListeners() m multicastEvent(ApplicationEvent) m multicastEvent(ApplicationEvent, ResolvableType) ApplicationEventPub ter | void void void void void void void void |

| ApplicationEventPublisher publishEvent |
|---|
| AbstractApplicationContext |
| protected void publishEvent [] Object |
| event,ResolvableType eventType[]{ |
| Assert.notNull event, "Event must not be null" ; |
| |
| if□this.earlyApplicationEvents !=null□{ |
| this.earlyApplicationEvents.add |
| <pre>□applicationEvent□;</pre> |
| } |
| else { |
| $get Application Event Multicaster \verb .multicast Event \\$ |
| <pre>□applicationEvent,eventType□;</pre> |
| } |
| |
| } |
| ApplicationEventMulticaster multicastEvent |
| |
| |

```
public void multicastEvent(final ApplicationEvent event, ResolvableType eventType) {
     ResolvableType type = (eventType != null ? eventType :
resolveDefaultEventType(event));
     for (final ApplicationListener<?> listener : getApplicationListeners(event, type)) {
       Executor executor = getTaskExecutor();
       if (executor != null) {
        executor.execute(new Runnable() {
          @Override
          public void run() {
            invokeListener(listener, event);
        });
       else {
        invokeListener(listener, event);
    protected void invokeListener (ApplicationListener listener, ApplicationEvent event) {
     ErrorHandler errorHandler = getErrorHandler();
     if (errorHandler != null) {
      try {
        listener.onApplicationEvent(event);
       catch (Throwable err) {
        errorHandler.handleError(err);
     }
     else {
      try {
       listener.onApplicationEvent(event);
       catch (ClassCastException ex) {
        LogFactory.getLog(getClass()).debug("Non-matching event type for listener: "
+ listener, ex);
       }
     }
```

| SimpleApplicationEventMulticaster | | | | | |
|--|-----|-----|--|--|--|
| ApplicationListener | Eve | nt∏ | | | |
| onApplicationEvent[][][][][][][][][][][][][] | | | | | |
| | | | | | |

```
@SuppressWarnings("serial")
@JsonTypeInfo(use = JsonTypeInfo.Id.NAME, property = "type")
@JsonIgnoreProperties("source")
public abstract class RemoteApplicationEvent extends ApplicationEvent {
   private static final Object TRANSIENT SOURCE = new Object();
   private final String originService;
   private final String destinationService;
   private final String id;
   protected RemoteApplicationEvent() {
      // for serialization libs like jackson
      this (TRANSIENT SOURCE, null, null);
   protected RemoteApplicationEvent(Object source, String originService,
         String destinationService) {
      super (source);
      this.originService = originService;
      if (destinationService == null) {
         destinationService = "*";
      if (!destinationService.contains(":")) {
         // All instances of the destination unless specifically requested
         destinationService = destinationService + ":**";
      this.destinationService = destinationService;
      this.id = UUID.randomUUID().toString();
   protected RemoteApplicationEvent(Object source, String originService) {
      this (source, originService, null);
```

□□□□RemoteApplicationEvent□□□□□□□□

```
●@JsonIgnoreProperties["source"[][][][][] source [][]
{
      "type": "RefreshRemoteApplicationEvent",
      "timestamp": 1475073160814,
      "originService": "config-server:7001",
      "destinationService": "*:**",
      "id": "bbfbf495-39d8-4ff9-93d6-174873ff7299"
   }
   {
      "type": "AckRemoteApplicationEvent",
      "timestamp": 1475075467554,
      "originService": "didispace:7002",
      "destinationService": "*:**",
      "id": "7560151e-f60c-49cd-8167-b691e846ad08".
                    "21502725-28f5-4d19-a98a-
      "ackId":
   f8114fa4f1dc",
      "ackDestinationService": "*:**",
      "event":
   "org.springframework.cloud.bus.event.RefreshRemoteA
   pplicationEvent"
```

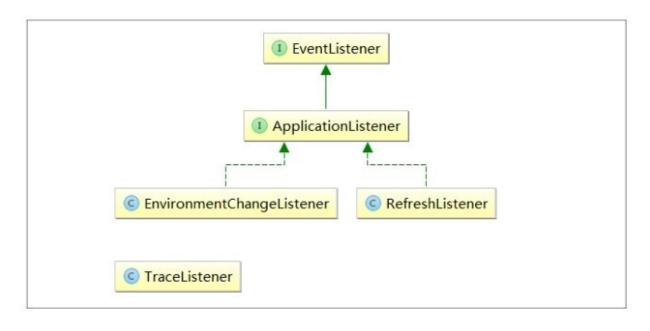
```
@SuppressWarnings[]"serial"[]
   public class RefreshRemoteApplicationEvent extends
 RemoteApplicationEvent {
    @SuppressWarnings[]"unused"[]
    private RefreshRemoteApplicationEvent□□{
     //for serializers
    }
          RefreshRemoteApplicationEvent |
    public
                                Object
  source, String origin Service,
        String destinationService [ {
     super\\\ source, origin\\ Service, destination\\ Service\\\;
    }
   }
  RefreshRemoteApplicationEvent [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] ackId []
RemoteApplicationEvent 
RemoteApplicationEvent
                    П
                        П
                            П
RefreshRemoteApplicationEvent[]
   @SuppressWarnings□"serial"□
             AckRemoteApplicationEvent
   public
        class
                               extends
 RemoteApplicationEvent {
```

```
private final String ackld;
        private final String ackDestinationService;
        private
                    final
                               Class<?
                                            extends
     RemoteApplicationEvent> event;
        @SuppressWarnings[]"unused"[]
        private AckRemoteApplicationEvent□□{
         super[];
         this.ackDestinationService=null;
         this.ackId=null;
         this.event=null:
        }
        public
                AckRemoteApplicationEvent ☐
                                             Object
     source, String originService,
             String
                              destinationService,String
          ackDestinationService,String ackld,
             Class<? extends RemoteApplicationEvent>
         type[[{
         super[]source,originService,destinationService[];
         this.ackDestinationService=ackDestinationServic
      e;
         this.ackId=ackId;
         this.event=type;
        }
```

```
@SuppressWarnings[]"serial"[]
     public
                                    class
  EnvironmentChangeRemoteApplicationEvent extends
   RemoteApplicationEvent {
      private final Map<String,String> values;
      @SuppressWarnings[]"unused"[]
      private
   EnvironmentChangeRemoteApplicationEvent□□{
       //for serializers
       values=null:
      }
      public EnvironmentChangeRemoteApplicationEvent
   □Object source, String originService,
          String
       destinationService,Map<String,String> values□{
       super\u00dfsource,originService,destinationService\u00df;
       this.values=values:
      }
  RemoteApplicationEvent>
Class<?
       extends
SentApplicationEvent
RemoteApplicationEvent
```

```
@SuppressWarnings[]"serial"[]
     @JsonTypeInfo
 ☐use=JsonTypeInfo.Id.NAME,property="type"☐
     @JsonIgnoreProperties | "source" |
                      SentApplicationEvent
    public
                                            extends
              class
 ApplicationEvent {
      private static final Object TRANSIENT_SOURCE=new
   Object∏∏;
      private final String originService;
      private final String destinationService;
      private final String id;
      private Class<? extends RemoteApplicationEvent>
   type;
      protected SentApplicationEvent∏∏{
        //for serialization libs like jackson
        this
    TRANSIENT SOURCE, null, null, null, Remote Application E
    vent.class∏;
      public SentApplicationEvent ☐ Object source, String
   originService,
           String destinationService, String id,
                             RemoteApplicationEvent>
           Class<? extends
        type∏{
```

```
super[]source[];
          this.originService=originService;
          this.type=type;
          if destination Service == null {
            destinationService="*";
          }
          if□! destinationService.contains□":"□□{
                 instances of the destination
            //All
                                               unless
        specifically requested
            destinationService=destinationService+":**";
          }
          this.destinationService=destinationService:
          this.id=id;
       }
   ____Spring Cloud Bus______
```



```
□□□ RefreshListener □ EnvironmentChangeListener □□□□
Spring [ ] [ ] [ ] [ ] [ ] [ ApplicationListener [ ] [ ] [ ] [ ] [
RefreshListener □
     public class RefreshListener
           implements
       ApplicationListener<RefreshRemoteApplicationEvent
       > {
         private
                              Log
                                     log=LogFactory.getLog
                    static
     □RefreshListener.class□;
         private ContextRefresher contextRefresher;
         public
                   RefreshListener
                                           ContextRefresher
                                      П
     contextRefresher∏{
           this.contextRefresher=contextRefresher:
         }
         @Override
         public
                          void
                                         onApplicationEvent
     □RefreshRemoteApplicationEvent event□{
```

```
Set<String> keys=contextRefresher.refresh□□;
           log.info [] "Received remote refresh request.Keys
       refreshed "+keys\;
         }
     }
   RefreshRemoteApplicationEvent [] onApplicationEvent
public class ContextRefresher {
         private ConfigurableApplicationContext context;
         public synchronized Set<String> refresh□□{
           Map<String,Object> before=extract□
                this.context.getEnvironment
             \square \square.getPropertySources\square \square \square;
           addConfigFilesToEnvironment□□;
           Set<String> keys=changes□before,
     extract
                     П
                               this.context.getEnvironment
 \square \square.getPropertySources\square \square \square \square.keySet\square \square;
           this.context.publishEvent
                                           П
                                                      new
       EnvironmentChangeEvent∏keys∏;
           this.scope.refreshAll | | ;
           return keys;
         }
       }
```

| Environme | entChange | Listener | | | |
|--|---|---|---------------|--------------------|----|
| public class I | Environme | ntChang | geListener | | |
| impleme | nts | | | | |
| ApplicationLi | stener <er< td=""><td>nvironme</td><td>entChangeRe</td><td>emoteAppli</td><td>ic</td></er<> | nvironme | entChangeRe | emoteAppli | ic |
| ationEvent> | { | | | | |
| private | static | Log | log=LogFa | ctory.getLo | og |
| <pre>[]Environment()</pre> | ChangeList | ener.cla | ss∏; | | |
| @Autowire | :d | | | | |
| private Env | vironment | Manager | env; | | |
| @Override | | | | | |
| public | VO | id | onAppl | icationEve | nt |
| <pre>□Environment(</pre> | ChangeRer | moteApp | licationEven | it event∐{ | |
| Map <str< td=""><td>ing,String</td><td>> values</td><td>=event.get\</td><td>/alues<u>□</u>□;</td><td></td></str<> | ing,String | > values | =event.get\ | /alues <u>□</u> □; | |
| log.info [|] "Receive | d remo | te environm | nent chang | ge |
| request.Keys | /values to | update | II | | |
| +val | ues[]; | | | | |
| for 🛮 | Map.En | try <strir< td=""><td>ng,String></td><td>entry</td><td>:</td></strir<> | ng,String> | entry | : |
| values.entry | Set <u></u> □□[{ | | | | |
| env.se | tProperty [|] entry.g | etKey 🛮 🗘 ,er | ıtry.getValı | ıе |
| | | | | | |
| } | | | | | |
| } | | | | | |
| } | | | | | |
| Environm | nentChang | eRemot | eApplication | Event [[[[[| |
| | |] [| | | |
| EnvironmentChang | | | | | |
|] Map | Environ | mentMa | nager[][][[[|][] | |
| | | | | | |

```
□□□□□□□RefreshListener□EnvironmentChangeListener□□□
public class TraceListener {
                                   log=LogFactory.getLog
       private
                  static
                           Log
   □TraceListener.class□;
       private TraceRepository repository;
       public TraceListener TraceRepository repository {
         this.repository=repository;
       }
       @EventListener
       public void onAck ☐ AckRemoteApplicationEvent
   event∏{
         this.repository.add | getReceived | Trace | event | | | ;
       @EventListener
       public void onSend∏SentApplicationEvent event∏{
         this.repository.add[getSentTrace[event[]];
       }
       protected
                    Map<String,Object>
                                            getSentTrace
   □SentApplicationEvent event□{
       }
                  Map<String,Object> getReceivedTrace
       protected
   □AckRemoteApplicationEvent event□{
       }
```

```
}
            _____ApplicationListener
_____ AckRemoteApplicationEvent _____
SentApplicationEvent□□□
            getReceivedTrace[event[]]; [] [] TraceRepository [] Trace[] [] []
InMemoryTraceRepository [ ] [ ] [ ]
                  public
                                                                InMemoryTraceRepository
                                                                                                                                                    implements
                                          class
      TraceRepository {
                        private int capacity=100;
                        private boolean reverse=true;
                                                               final
                                                                                              List<Trace>
                        private
                                                                                                                                                    traces=new
            LinkedList<Trace>□□;
                        public void setReverse∏boolean reverse∏{
                              synchronized ☐this.traces ☐ {
                                   this.reverse=reverse;
                              }
                        }
                        public void setCapacity□int capacity□{
                              synchronized \text{\text{this.traces} \text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\ti}\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text
                                   this.capacity=capacity;
                              }
                        @Override
```

```
public List<Trace> findAll□□{
       synchronized ☐this.traces ☐ {
                Collections.unmodifiableList
         return
                                         П
                                            new
      ArrayList<Trace> □ this.traces □ □;
       }
      }
      @Override
      public void add Map < String, Object > map {
       Trace trace=new Trace new Date n,map;
       synchronized this.traces {
         while | this.traces.size | | >= this.capacity | {
          this.traces.remove | this.reverse ? this.capacity-
       1:0 \sqcap;
         }
         if∏this.reverse∏{
          this.traces.add \( \)0,trace \( \);
         }
         else {
          this.traces.add∏trace∏;
       }
   Map<String,Object> map
```

```
getReceivedTrace [ ] [ ] [ ] [
     public class TraceListener {
       protected
                                            getSentTrace
                    Map<String,Object>
   □SentApplicationEvent event□{
         Map<String,Object>
                                               map=new
     LinkedHashMap<String,Object>□□;
         map.put□"signal","spring.cloud.bus.sent"□;
         map.put | "type",event.getType | | .getSimpleName
     map.put∏"id",event.getId∏∏;
         map.put["origin",event.getOriginService[[[]];
         map.put\(\pi\)"destination",event.getDestinationService
     \Pi\Pi\Pi;
         if□log.isDebugEnabled□□□{
           log.debug∏map∏;
         }
         return map;
                                        getReceivedTrace
       protected
                  Map<String,Object>
   □AckRemoteApplicationEvent event□{
         Map<String,Object>
                                               map=new
     LinkedHashMap<String,Object>□□;
         map.put□"signal","spring.cloud.bus.ack"□;
                                   "event", event.getEvent
         map.put
                         П
     \square\square.getSimpleName\square\square;
```

```
map.put[]"id",event.getAckId[][];
      map.put[]"origin",event.getOriginService[][];
      map.put
   □"destination",event.getAckDestinationService□□□;
      if□log.isDebugEnabled□□□{
       log.debug∏map∏;
      return map;
     }
  ____Spring Cloud Bus____ Trace ______
spring.cloud.bus.trace.enabled=true
  {
      "timestamp": 1475129670494,
      "info": {
       "signal": "spring.cloud.bus.ack",
       "event": "RefreshRemoteApplicationEvent",
       "id": "84ecdf83-a904-41bc-a34d-62680ccf35d7",
       "origin": "config-server:7001",
       "destination": "*:**"
      }
```

```
},
                             {
                                   "timestamp": 1475129670475,
                                    "info": {
                                          "signal": "spring.cloud.bus.sent",
                                          "type": "RefreshRemoteApplicationEvent",
                                          "id": "84ecdf83-a904-41bc-a34d-62680ccf35d7",
                                          "origin": "config-server:7001",
                                          "destination": "*:**"
                                    }
                            },{
                                    "timestamp": 1475129670473,
                                    "info": {
                                          "signal": "spring.cloud.bus.ack",
                                          "event": "RefreshRemoteApplicationEvent",
                                          "id": "84ecdf83-a904-41bc-a34d-62680ccf35d7",
                                          "origin": "didispace:7002",
                                          "destination": "*:**"
                                    }
                            }
                      1
                                     AckRemoteApplicationEvent
SentApplicationEvent
ONDO DE LA CIENTA DEL CIENTA DE LA CIENTA DEL CIENTA DE LA CIENTA DEL CIENTA DEL CIENTA DEL CIENTA DE LA CIENTA DEL CIENTA D
```

```
org.springframework.cloud.bus
Cloud Bus
@Configuration
    @ConditionalOnBusEnabled
    @EnableBinding[SpringCloudBusClient.class[]
    @EnableConfigurationProperties ☐ BusProperties.class ☐
    public
                 BusAutoConfiguration
                                  implements
           class
 ApplicationEventPublisherAware {
     public
                            final
                                      String
                 static
  BUS PATH MATCHER NAME="busPathMatcher";
     @Autowired
     @Output∏SpringCloudBusClient.OUTPUT∏
     private MessageChannel cloudBusOutboundChannel;
     @Autowired
     private ServiceMatcher serviceMatcher;
     @Autowired
     private ChannelBindingServiceProperties bindings;
     @Autowired
     private BusProperties bus;
                        ApplicationEventPublisher
     private
  applicationEventPublisher;
```

. . .

```
}
   ■MessageChannel cloudBusOutboundChannel
public
         boolean isFromSelf | RemoteApplicationEvent
 event∏{
      String originService=event.getOriginService□□;
     String serviceId=getServiceId□□;
      return this.matcher.match□originService,serviceId□;
    }
    public
          boolean isForSelf ☐ RemoteApplicationEvent
 event∏{
     String
   destinationService=event.getDestinationService□□;
                    destinationService==null
      return
   destinationService.trim[].isEmpty[][][this.matcher.match
   \sqcapdestinationService,getServiceId\sqcap\sqcap\sqcap\sqcap;
    }
   ■BusProperties bus☐☐☐☐☐☐☐Spring Cloud Bus☐☐☐☐☐☐☐☐
@ConfigurationProperties□"spring.cloud.bus"□
    public class BusProperties {
      private Env env=new Env□□;
```

```
private Refresh refresh=new Refresh□□;
                        private Ack ack=new Ack□□;
                        private Trace trace=new Trace∏;
                        private String destination="springCloudBus";
                        private boolean enabled=true;
            ____Spring Cloud Bus _____ spring.cloud.bus_
חחחחחחחח spring.cloud.bus.enabled
            Env Refresh Ack Refresh Trace Refresh 
"□□□□"□□□□□spring.cloud.bus.trace.enabled=true□□□□□
                 public static class Env {
                             private boolean enabled=true;
                 public static class Refresh {
                             private boolean enabled=true;
                 public static class Ack {
                             private boolean enabled=true;
                             private String destinationService;
```

```
}
  public static class Trace {
    private boolean enabled=false;
  }
 ______acceptLocal_
acceptRemote □
 RemoteApplicationEvent
@EventListener
 □classes=RemoteApplicationEvent.class□
   public void acceptLocal [] RemoteApplicationEvent
 event∏{
    if □this.serviceMatcher.isFromSelf □event □
       &&
                       instanceof
              П
                  event
     AckRemoteApplicationEvent□□{
     this.cloudBusOutboundChannel.send
   ☐MessageBuilder.withPayload☐event☐.
  build∏∏;
   }
```

```
SpringCloudBusClient.INPUT
___@EnableBinding _______
Cloud Stream
@StreamListener Spring Cloud Bus Client. INPUT
    public void acceptRemote 

☐ RemoteApplicationEvent
 event∏{
     if∏event instanceof AckRemoteApplicationEvent∏{
       if \sqcap this.bus.getTrace \sqcap \sqcap .isEnabled \sqcap \sqcap &&!
    this.serviceMatcher.isFromSelf[]event[]
         && this.applicationEventPublisher !=null∏{
        this.applicationEventPublisher.publishEvent
     ∏event∏;
       }//If it's an ACK we are finished processing at this
    point
       return;
     }
     if□this.serviceMatcher.isForSelf□event□
        && this.applicationEventPublisher !=null [ {
       if∏! this.serviceMatcher.isFromSelf∏event∏∏{
        this.applicationEventPublisher.publishEvent
     ∏event∏;
       }
       if∏this.bus.getAck∏∏.isEnabled∏∏∏{
```

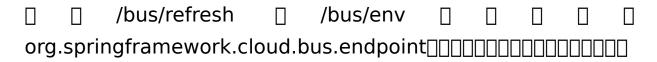
```
AckRemoteApplicationEvent

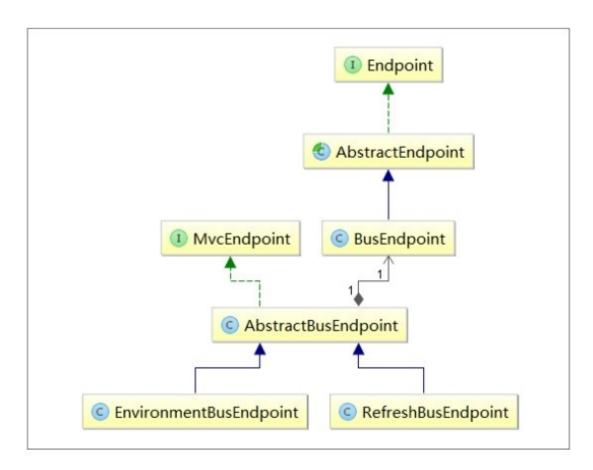
☐this,
            this.serviceMatcher.getServiceId∏,
            this.bus.getAck ... getDestinationService ...,
            event.getDestinationService [ ] ,event.getId
         \square\square, event.getClass\square\square\square;
           this.cloudBusOutboundChannel
             .send ☐ MessageBuilder.withPayload ☐ ack ☐ .build
         this.applicationEventPublisher.publishEvent
       ∏ack∏;
         }
          \sqcap this.bus.getTrace \sqcap \sqcap .isEnabled \sqcap \sqcap
                                                      &&
   this.applicationEventPublisher !=null∏{
         //We are set to register sent events so publish it for
     local consumption,
         //irrespective of the origin
         this.applicationEventPublisher.publishEvent ☐ new
     SentApplicationEvent∏this,
            event.getOriginService
         \square \square, event.getDestinationService \square \square,
             event.getId∏,event.getClass∏∏∏;
       }
```

AckRemoteApplicationEvent

ack=new

| $\verb $ |
|---|
| AckRemoteApplicationEvent |
| |
| Spring Cloud Bus |
| |
| 00000000000000000000000000000000000000 |
| <code>_/bus/refresh</code> |
| |
| 2016-09-30 11:05:13.037 INFO 18720[|
| main] |
| o.s.b.a.e.mvc.EndpointHandlerMapping : Mapped " |
| {[/bus/refresh],methods=[POST]}" |
| onto public void |
| org.springframework.cloud.bus.endpoint.RefreshBusEn |
| dpoint.refresh□java.lang.String□ |
| 2016-09-30 11:05:13.045 INFO 18720[|
| main] |
| o.s.b.a.e.mvc.EndpointHandlerMapping : Mapped " |
| {[/bus/env],methods=[POST]}" onto |
| public void |
| org.springframework.cloud.bus.endpoint.EnvironmentB |
| usEndpoint.env□java.util.Map <jav< td=""></jav<> |
| a.lang.String,java.lang.String>,java.lang.String[] |
| |
| org.springframework.cloud.bus.endpoint \Box \Box |
| $RefreshBusEndpoint \verb EnvironmentBusEndpoint \verb $ |





_____Spring Cloud Bus__Endpoint____spring-boot-actuator_____spring-boot-actuator_____

```
■ BusEndpoint [] [] [] [] AbstractEndPoint [] [] [] [] []
@ConfigurationProperties [][][][][]Spring Cloud Bus [][][][][]
_____id_bus______true_
    @ConfigurationProperties
 □prefix="endpoints.bus",ignoreUnknownFields=false□
     public
                class
                           BusEndpoint
                                            extends
 AbstractEndpoint<Collection<String>> {
      public BusEndpoint∏∏{
        super[]"bus"[];
      }
      @Override
      public Collection<String> invoke∏{
        return Collections.emptyList∏;
      }
     }
    public abstract class AbstractEndpoint<T> implements
 Endpoint<T>,EnvironmentAware {
      this id, true;
      public AbstractEndpoint ☐ String id, boolean sensitive ☐
   {
        this.id=id;
        this.sensitiveDefault=sensitive:
      }
```

```
}
   BusEndpoint□□□□□□□□□□□Environment□□□□□
    public
            class
                  AbstractBusEndpoint
                                     implements
 MvcEndpoint {
      private ApplicationEventPublisher context;
      private BusEndpoint delegate;
      private String appld;
      public
                              AbstractBusEndpoint
   □ApplicationEventPublisher context,String appld,
          BusEndpoint busEndpoint∏{
       this.context=context:
       this.appld=appld;
       this.delegate=busEndpoint;
      }
      protected String getInstanceId∏∏{
       return this.appld;
      }
      protected void publish Application Event event {
       context.publishEvent∏event∏;
      }
      @Override
      public String getPath∏∏{
       return "/"+this.delegate.getId□□;
```

```
}
                                          @Override
                                         public boolean isSensitive∏∏{
                                                   return this.delegate.isSensitive∏;
                                         }
                                         @Override
                                         @SuppressWarnings[]"rawtypes"[]
                                         public Class<? extends Endpoint> getEndpointType
                    \Pi\Pi{
                                                   return this.delegate.getClass□□;
                                         }
                               }
                     □□□□□□□□□□□ AbstractBusEndpoint □□□□ MVC □□□□□□□
 nefresh nefres
AbstractBusEndpoint | getPath | for the stractBusEndpoint | for the stractBusEndpoin
☐ RefreshBusEndpoint ☐ refresh ☐☐☐☐☐☐☐/bus/refresh☐☐☐☐☐
_____publish ___
 RefreshRemoteApplicationEvent
                                                                                                                                                  RefreshBusEndpoint
                                         public
                                                                                                class
                                                                                                                                                                                                                                                                                 extends
                     AbstractBusEndpoint {
                                                   public
                                                                                                                                                                                                                    RefreshBusEndpoint
                              □ApplicationEventPublisher context,String id,
                                                                        BusEndpoint delegate∏{
                                                             super context, id, delegate;
```

```
}
       @RequestMapping
    □value="refresh",method=RequestMethod.POST□
       @ResponseBody
       public void refresh□
           @RequestParam
          value="destination",required=false [
                                           String
       destination∏{
         publish ☐ new RefreshRemoteApplicationEvent
      \sqcapthis,getInstanceId\sqcap\sqcap,
    destination∏;
        }
      }
   ●EnvironmentBusEndpoint□□□□RefreshBusEndpoint□□□□
public
                   EnvironmentBusEndpoint
            class
                                          extends
 AbstractBusEndpoint {
       public
                            EnvironmentBusEndpoint
    □ApplicationEventPublisher context,String id,
            BusEndpoint delegate [ {
         super\\\context,id,delegate\\\;
        }
       @RequestMapping
    □value="env",method=RequestMethod.POST□
       @ResponseBody
```

| public void | env | | @Requ | estParam |
|--|-------------|------------------------------|--------------|----------------------------------|
| Map <string,string></string,string> | params, | | | |
| @Request | Param | | | |
| □ value="des | tination",r | equired | =false | ☐ String |
| destination[[{ | | | | |
| publish | | | | new |
| EnvironmentChang | geRemote/ | Applicat | ionEvent | <u>-</u> |
| | d□□, | | | |
| destinati | ion,param | s∏ <u></u> ; | | |
| } | | | | |
| } | | | | |
| | | l | | |
| Spring Cloud E
Spring Cloud E
Spring Cloud Bus | | bbitMQ
3000000
3000000 | Kafka [
 | 10000000
10000000
10000000 |
| | | nqp∏spı | ing-clou | d-starter- |
| bus-kafka <u>□</u> □□□ | | | | |
| 🛮 🗎 🗎 🗎 🗎 spring-clou | d-starter-k | ous-amo | p 🛮 spriι | ng-cloud- |
| starter bus-kafka 🛮 🗎 🗎 🖺 🖺 | 0000000 |][] spi | ring-clou | d-starter- |
| streamrabbit 🛮 spring-clou | d-starter-s | stream-l | kafka∏∏ | |
| Spring Cloud | Stream 🛚 [| | | □□Spring |
| Cloud Bus | Spring Clo | ud Strea | am[][][[| |
| □□□□Spring Cloud Bus□□□□
□□□□Spring Cloud Stream□ | | | | m |

| Binder |
|-------------------|
| RabbitMQ_Kafka |
| RabbitMQ |
| BSpring Cloud Bus |
| |

10 Cloud Stream

Spring Cloud Stream Spring Spring Spring Boot Spring Integration Cloud Stream Spring Boot Spring Cloud Stream Spring Boot Spring Cloud Stream Cloud

- ■RabbitMQ
- ●Kafka



```
_____Spring Boot_______
RabbitMQ
   ●□□□□□□Spring Boot□□□□□stream-hello□
   ● □ □ pom.xml □ □ □ □ □ □ □ Spring Cloud Stream □ RabbitMQ □ □
<parent>
       <groupId>org.springframework.boot</groupId>
       <artifactId>spring-boot-starter-parent</artifactId>
       <version>1.3.7.RELEASE
       <relativePath/> <!--lookup parent from repository-
    ->
    </parent>
    <dependencies>
       <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
       </dependency>
       <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
       </dependency>
       <dependency>
        <groupId>org.springframework.cloud</groupId>
```

```
<artifactId>spring-cloud-starter-stream-
   rabbit</artifactId>
     </dependency>
 </dependencies>
 <dependencyManagement>
     <dependencies>
       <dependency>
          <groupId>org.springframework.cloud
       pld>
          <artifactId>spring-cloud-
       dependencies</artifactId>
           <version>Brixton.SR5</version>
           <type>pom</type>
           <scope>import</scope>
       </dependency>
     </dependencies>
 </dependencyManagement>
●□□□□□□□RabbitMQ□□□□□SinkReceiver□□□□□□
 @EnableBinding ☐Sink.class ☐
 public class SinkReceiver {
     private
                          static
                                              Logger
 logger=LoggerFactory.getLogger
 □HelloApplication.class□;
     @StreamListener[]Sink.INPUT[]
     public void receive \ Object payload \ {
       logger.info∏"Received: "+payload∏;
     }
```

```
}
   @SpringBootApplication
    public class HelloApplication {
        public static void main[]String[]args[]{
         SpringApplication.run
      ☐HelloApplication.class,args☐;
        }
    }
   ____RabbitMQ ___Spring
Boot
   ●□□□□□Spring Boot□□□□□□□
      INFO
                                          16272---
                                          :declarin
   [main]o.s.c.s.b.r.RabbitMessageChannelBinder
   g queue for
    inbound:
 input.anonymous.Y8VsFILmSC27eS5StsXp6A,bound
                                               to:
 input
      INFO
                                          16272---
   [main]o.s.a.r.c.CachingConnectionFactory
                                         : Created
   new
    connection:
 SimpleConnection@3c78e551[delegate=amqp://guest@12
 7.0.0.1:5672/]
      INFO
                                          16272---
                                         : Channel
   [main]o.s.integration.channel.DirectChannel
```

'input.anonymous.Y8VsFILmSC27eS5StsXp6A.bridge' has 1 subscriber[]s[].

INFO 16272---

[main]o.s.i.a.i.AmqpInboundChannelAdapter : started inbound.input.anonymous.Y8VsFILmSC27eS5StsXp6A

. . .







| Delivery mode: | | v Victorial Rey Impacta | nonymous.Y8VsFILmSC27eS5StsXp6A, routing it to | cino que |
|-----------------|------------------|-------------------------|--|----------|
| | 1 Non persistent | | | |
| Headers: (?) | | = | String • | |
| Properties: (?) | | = | | |
| | | | | |
| | | | | |

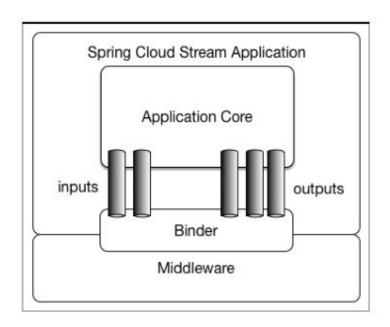
| Spring Boot | |
|--|-----------------|
| INFO 16272[C2 | 7eS5StsXp6A- |
| 1]com.didispace.HelloApplication :Rece | ived: |
| [B@7cba610e | |
| | ve [][[][] |
| | 30000000000 |
| | |
| | Spring Boot∏∏ |
| | |
| <pre>Description</pre> <pre>Description</pre> <pre>Description</pre> <pre>Spring</pre> <pre>Boot</pre> <pre>Description</pre> <pre>Description</pre> <pre>Spring</pre> <pre>Boot</pre> <pre>Description</pre> <pre>Description</pre> <pre>Spring</pre> <pre>Boot</pre> <pre>Description</pre> <pre>Description</pre> <pre>Spring</pre> <pre>Spring</pre> <pre>Description</pre> | -cloud-starter- |
| streamrabbit | obitMQ |
| DDDDDRabbitMQ DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD | 0000000000 |
| spring-cloud-stream-binder-rabbit□□□ | |
| <dependencies></dependencies> | |
| <dependency></dependency> | |
| <pre><groupid>org.springframework.clo</groupid></pre> | ud |
| | -
- |

```
<artifactId>spring-cloud-stream-binder-
             rabbit</artifactId>
                </dependency>
          </dependencies>
      Ondoor Stream
SinkReceiver□□
      _____Channel ______
@EnableBinding∏Sink.class∏∏∏ Sink ∏∏∏∏∏∏ Spring Cloud
Stream
         public interface Sink {
                String INPUT="input";
                @Input[Sink.INPUT]
                SubscribableChannel input∏;
      Description:

Stream [][[][][] output [][] Source [][][][][] Sink [Source]
Processor_____@Output______
_____@EnableBinding __________________
@EnableBinding□value={Sink.class,Source.class}□□
      Sink.INPUT
RabbitMQnnnnnnnnnnnreceivennnnnnnnnnn
```

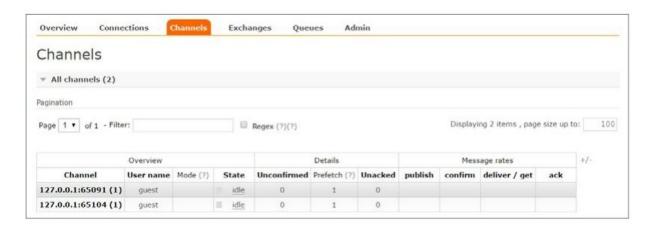


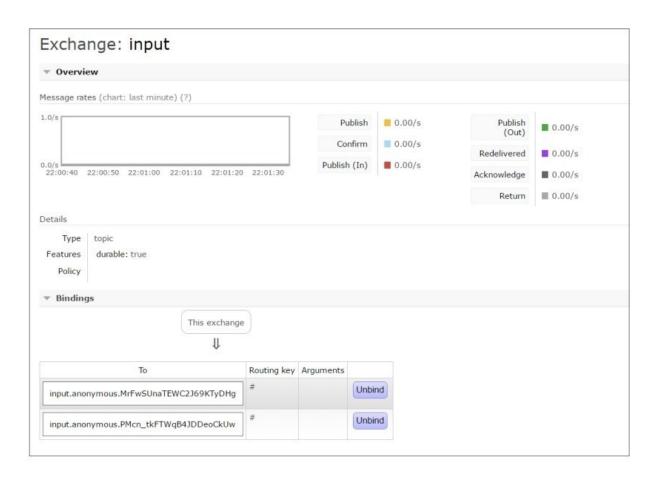
Cloud Stream



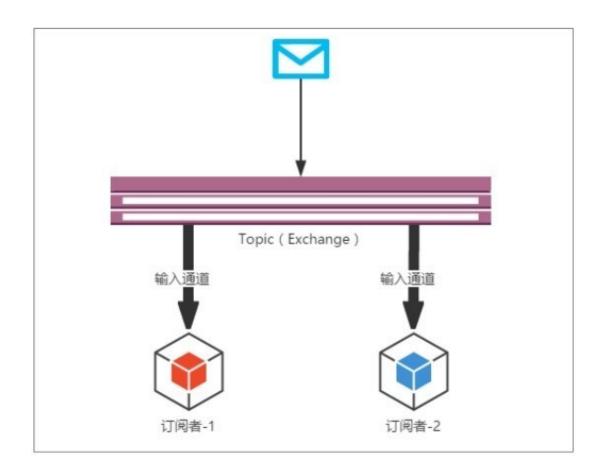
| _Channel |
|---|
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| $Boot \verb $ |
| |
| |
| Binder RabbitMQ Binder |
| Spring Cloud Stream |
| RabbitMQ |
| Binder |
| application.properties |
| $\cite{Annext} \begin{tabular}{lllllllllllllllllllllllllllllllllll$ |
| $\verb $ |
| <pre></pre> |
| $application.yml \verb $ |
| input |
| spring.cloud.stream.bindings.input.destination=raw- |
| sensor-data |
| spring.rabbitmq.host=localhost |
| spring.rabbitmq.port=5672 |
| spring.rabbitmq.username=springcloud |
| spring.rabbitmq.password=123456 |

| Spring Cloud Stream |
|-------------------------------|
| |
| Topic Spring Cloud Stream |
| RabbitMQ |
| Exchange |
| |

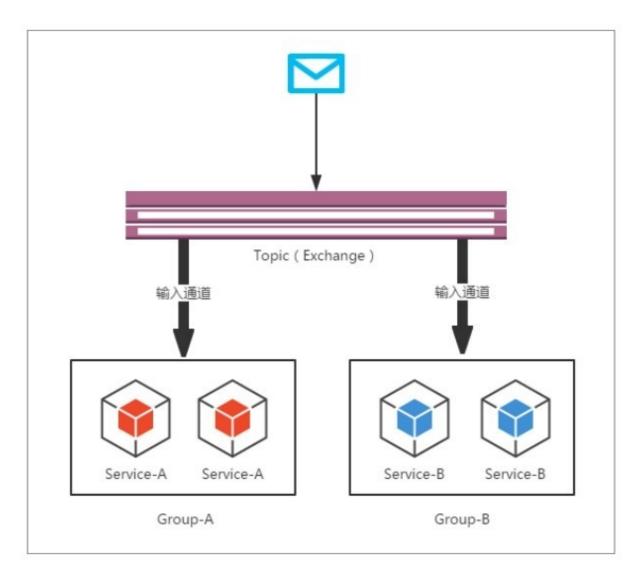








 Group-A
 Group-B
 Group-B



| Stream |
|--------|
| |
| |
| |
| |
| |
| |

```
@Target
 □{ElementType.TYPE,ElementType.ANNOTATION TYPE}□
   @Retention | Retention Policy. RUNTIME |
   @Documented
   @Inherited
   @Configuration
   @Import[]{
     ChannelBindingServiceConfiguration.class,
     BindingBeansRegistrar.class,
     BinderFactoryConfiguration.class,
     @EnableIntegration
   public @interface EnableBinding {
      Class<? >[]value∏default {};
  MessageConverterConfigurer
                    П
                       П
                          П
                             П
BindableChannelFactory
BindingBeansRegistrar
                         П
                                  П
ImportBeanDefinitionRegistrar
\sqcap \square \square Bean \sqcap \sqcap \sqcap \square BindingBeansRegistrar \sqcap
```

| $\verb $ |
|--|
| Description of the control of the co |
| BeanSink |
| ◆BinderFactoryConfiguration:Binder□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| SpelExpressionConverterConfiguration:SpEL |
| @EnableBinding |
| @Import_BindingBeansRegistrar |
| value |
| |
| |
| |
| |
| Spring Cloud Steam |
| Spring Cloud Steam @DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| |
| |
| |
| |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| <pre>Describable Binding Describe Bindin</pre> |
| <pre>DescribableChannel input</pre> <pre>EnableBinding</pre> <pre>Processor</pre> <pre>DescribableChannel input</pre> <pre>EnableBinding</pre> <pre>Procel</pre> <pre>Processor</pre> <pre>DescribableChannel input</pre> <pre>DescribableChannel</pre> <pre>EnableBinding</pre> <pre>Processor</pre> <pre>DescribableChannel</pre> <pre>DescribableChannel</pre> <pre>EnableBinding</pre> <pre>DescribableChannel</pre> <pre>DescribableC</pre> |

```
@Output[]Source.OUTPUT[]
                  MessageChannel output□□;
              }
              public interface Processor extends Source,Sink {
              }
         ______Sink_Source_____@Input_@Output___
nnnnnnnvaluennnnnnnnnnnnnnnn
         ПППППППППППППППППППППППППППППППSubscribableChannelП
On the stream of the stream of
input
         public interface SinkSender {
                       @Output∏Sink.INPUT∏
                       MessageChannel output∏;
         SinkSender□□□□□□□Spring Cloud Stream□□□□□□□□□
              @EnableBinding[value={Sink.class,SinkSender.class}[]
```

```
public class SinkReceiver {
      private
                         static
                                          Logger
   logger=LoggerFactory.getLogger∏SinkReceiver.class∏;
      @StreamListener∏Sink.INPUT∏
      public void receive Object payload { {
       logger.info∏"Received: "+payload∏;
      }
    }
   @RunWith∏SpringJUnit4ClassRunner.class∏
    @SpringApplicationConfiguration
 □classes=HelloApplication.class□
    @WebAppConfiguration
    public class HelloApplicationTests {
      @Autowired
      private SinkSender sinkSender;
      @Test
      public void contextLoads∏{
       sinkSender.output
                                           .send
                                    П
          MessageBuilder.withPayload
                                     П
                                           "From
    SinkSender"∏.build∏∏;
      }
   ullet
____input_____
```

. . .

```
INFO
          10656---[
  main]com.didispace.HelloApplication
                          : Received:
   From SinkSender
  □□Spring Cloud Stream□□□□□□□□□@Input□@Output□□□□□
□□□□SinkSender□□□□□□□□□□input□□□□□□□
   @RunWith□SpringJUnit4ClassRunner.class□
   @SpringApplicationConfiguration
 □classes=HelloApplication.class□
   @WebAppConfiguration
   public class HelloApplicationTests {
    @Autowired
    private MessageChannel input;
    @Test
    public void contextLoads∏{
     input.send [] MessageBuilder.withPayload [] "From
   MessageChannel"□.build□□□;
    }
   }
  _____@Qualifier_______
```

| MessageChannel @Output |
|--|
| Output-1 |
| 1@Qualifier_"Output-1" |
| public interface MySource { |
| @Output□"Output-1"□ |
| MessageChannel output1□□; |
| @Output□"Output-2"□ |
| MessageChannel output2□□; |
| } |
| @Component |
| <pre>public class OutputSender {</pre> |
| @Autowired @Qualifier[]"Output-1"[] |
| private MessageChannel output; |
| ••• |
| } |
| |
| ☐☐ Spring Cloud Stream ☐☐☐ Spring Integration ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ |
| Spring Cloud Stream DOD Spring |
| Integration DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Cloud Stream DDD@StreamListener |
| |
| Spring Integration 🗆 🗆 🗆 |
| input |
| <pre>Description</pre> |
| @InboundChannelAdapter |
| |

```
@EnableBinding[value={Sink.class}[]
   public class SinkReceiver {
     private
                           static
                                                Logger
 logger=LoggerFactory.getLogger[|SinkReceiver.class[];
     @ServiceActivator inputChannel = Sink. INPUT
     public void receive \ Object payload \ {
       logger.info□"Received: "+payload□;
     }
   }
   @EnableBinding
                                               value=
                                П
public class SinkSender {
     private
                           static
                                                Logger
 logger=LoggerFactory.getLogger∏SinkSender.class∏;
     @Bean
     @InboundChannelAdapter
               value=SinkOutput.OUTPUT,poller=@Poller
 П
 □fixedDelay="2000"□□
     public MessageSource<Date> timerMessageSource
 \Pi\Pi{
       return --> new GenericMessage <> new Date --;
     public interface SinkOutput {
       String OUTPUT="input";
       @Output∏SinkOutput.OUTPUT∏
       MessageChannel output∏;
     }
```

} @ServiceActivator [][][][]@StreamListener[][][][]Sink.INPUT[][] _____SinkSender ______ @InboundChannelAdapter **INFO** 248---[ask-scheduler-2]com.didispace.HelloApplication : Received: Sun Nov 13 11:22:39 CST 2016 248---[ask-scheduler-INFO 3]com.didispace.HelloApplication : Received: Sun Nov 13 11:22:41 CST 2016 248---[ask-scheduler-**INFO** 2]com.didispace.HelloApplication : Received: Sun Nov 13 11:22:43 CST 2016 248---[ask-scheduler-INFO 1]com.didispace.HelloApplication : Received: Sun Nov 13 11:22:45 CST 2016 248---[ask-scheduler-**INFO** 5]com.didispace.HelloApplication : Received: Sun Nov 13 11:22:47 CST 2016

```
SinkSender
    @Transformer
 □inputChannel=Sink.INPUT,outputChannel=Sink.INPUT
    public Object transform[Date message[]{
                   SimpleDateFormat □ "yyyy-MM-dd
      return
             new
   HH:mm:ss"∏.format∏message∏;
    }
   nnnnnnnnnnnnnnnnnnnnnnnyyyy-MM-dd
HH:mm:ss□□□□
      INFO
                              22092---[ask-scheduler-
   2]com.didispace.HelloApplication
                                    :Received:
    2016-11-13 12:12:21
      INFO
                              22092---[ask-scheduler-
   3]com.didispace.HelloApplication
                                    :Received:
    2016-11-13 12:12:23
                              22092---[ask-scheduler-
      INFO
                                    :Received:
   2]com.didispace.HelloApplication
    2016-11-13 12:12:25
      INFO
                              22092---[ask-scheduler-
                                    :Received:
   1]com.didispace.HelloApplication
    2016-11-13 12:12:27
      INFO
                              22092---[ask-scheduler-
   5]com.didispace.HelloApplication
                                    :Received:
    2016-11-13 12:12:29
   @StreamListener []
```

```
________StreamListener_________
On One of the American Cloud Steam
@ServiceActivator
   @EnableBinding□value={Sink.class}□
   public class SinkReceiver {
    private
                static
                           Logger
  logger=LoggerFactory.getLogger[|SinkReceiver.class[];
    @ServiceActivator∏inputChannel=Sink.INPUT∏
    public void receive User user {
     logger.info□"Received: "+user□;
    }
    @Transformer
  □inputChannel=Sink.INPUT,outputChannel=Sink.INPUT□
    public User transform ☐ String message ☐ throws
  Exception {
     ObjectMapper objectMapper=new ObjectMapper
   \Pi\Pi;
```

```
user=objectMapper.readValue
         User
     □message,User.class□;
         return user;
       }
     }
       @EnableBinding
                                                 value=
                                   П
   {SinkSender.SinkOutput.class}[]
       public class SinkSender {
       private
                             static
                                                 Logger
   logger=LoggerFactory.getLogger[|SinkSender.class[];
       @Bean
       @InboundChannelAdapter
   □value=Sink.INPUT,poller=@Poller□fixedDelay="2000"□□
       public MessageSource<String> timerMessageSource
   \Pi\Pi{
         return □
                  \sqcap -> new
                                GenericMessage<>
     {\model} \ {\"name\":\"didi\",\"age\":30}"\;
       }
       public interface SinkOutput {
         String OUTPUT="input";
         @Output[|SinkOutput.OUTPUT[]
         MessageChannel output□□;
       }

    @ServiceActivator [] [] [] [] [] [] [] [] [] [] User [] []
```

```
@EnableBinding□value={Sink.class}□
  public class SinkReceiver {
   private
               static
                         Logger
 logger=LoggerFactory.getLogger[|SinkReceiver.class[];
   @StreamListener∏Sink.INPUT∏
   public void receive∏User user∏{
    logger.info□"Received: "+user□;
   }
  }
 _____@Transformer ______
@StreamListener□□□@ServiceActivator□□□□□SinkSender□□□□□
            П
              П
                П
                  П
                   П
spring.cloud.stream.bindings.input.contenttype=application/j
@StreamListener
Spring Cloud Steam
______input ____content-type
____App1_App2_
```

```
@EnableBinding[value={Processor.class}[]
   public class App1 {
      private
                                Logger
                   static
   logger=LoggerFactory.getLogger[App1.class];
      @StreamListener □ Processor. INPUT □
      @SendTo∏Processor.OUTPUT∏
      public Object receiveFromInput∏Object payload∏{
       logger.info□"Received: "+payload□;
       return "From Input Channel Return-"+payload;
      }
   }
  ____App2____input____output____output
____input_____ App1_
spring.cloud.stream.bindings.input.destination=output
   spring.cloud.stream.bindings.output.destination=input
  _____App1__input
@EnableBinding□value={Processor.class}□
   public class App2 {
      private
                   static
                                Logger
   logger=LoggerFactory.getLogger∏App2.class∏;
      @Bean
```

```
@InboundChannelAdapter
   Dvalue=Processor.OUTPUT,poller=@PollerDfixedDelay=
  "2000"□□
      public
                        MessageSource<Date>
   timerMessageSource∏{
       return □ □ -> new GenericMessage <> □ new Date
     \Pi\Pi\Pi;
      }
      public void receiveFromOutput∏Object payload∏{
       logger.info∏"Received: "+payload∏;
      }
    }
  INFO
                     28180---[mWYODqFPjxMmg-
  1]com.didispace.HelloApplication : Received:
   Wed Nov 16 08:19:16 CST 2016
     INFO
                     28180---[mWYODgFPixMmg-
  1]com.didispace.HelloApplication : Received:
   Wed Nov 16 08:19:18 CST 2016
     INFO
                     28180---[mWYODqFPjxMmg-
  1]com.didispace.HelloApplication
                           : Received:
   Wed Nov 16 08:19:20 CST 2016
```

```
31916---[xmrMN07yhd8Ag-
      INFO
   1]com.didispace.HelloApplication
                               : Received:
    From Input Channel Return-Wed Nov 16 08:19:16 CST
 2016
                          31916---[xmrMN07yhd8Ag-
      INFO
   1]com.didispace.HelloApplication
                                : Received:
    From Input Channel Return-Wed Nov 16 08:19:18 CST
 2016
      INFO
                          31916---[xmrMN07yhd8Ag-
   1]com.didispace.HelloApplication
                                : Received:
    From Input Channel Return-Wed Nov 16 08:19:20 CST
 2016
   @EnableScheduling
    @EnableBinding \textsqrapprox value = { Processor.class } \tag{}
    public class App1 {
       private
                         static
                                         Logger
    logger=LoggerFactory.getLogger∏App1.class∏;
       @ServiceActivator
    □ inputChannel=Processor.INPUT,outputChannel=Proce
    ssor.OUTPUT∏
       public Object receiveFromInput∏Object payload∏{
         logger.info□"Received: "+payload□;
```

```
}
    }
                    Spring Cloud Stream
__App1____App2____input______output____
App2
   ● pom.xml pom.xml spring-cloud-stream-rxjava pom.xml
    <dependency>
       <groupId>org.springframework.cloud</groupId>
       <artifactId>spring-cloud-stream-
    rxjava</artifactId>
       <version>1.0.2.RELEASE
    </dependency>
   ●□□App1□□□□□□□□
    @EnableRxJavaProcessor
    public class App1 {
      private
                                         Logger
                        static
   logger=LoggerFactory.getLogger[App1.class];
      @Bean
      public RxJavaProcessor<String,String> processor□□{
       return inputStream-> inputStream.map

☐data-> {
         logger.info□"Received: "+data□;
         return data;
```

return "From Input Channel Return-"+payload;

```
} ☐ .map ☐ data-> String.valueOf ☐ "From Input
              Channel Return-"+data∏;
                    }
               }
          ● □ App1 □ □ □ □ □ □ □ @ Enable Rx Java Processor □ □ □ □ □ □
Processor [ [ [ [ [ [ [ Processor.class ] [ [ [ Processor.class ] [ [ [ Processor.class ] [ [ [ [ Processor.class ] [ 
               @Target∏{ElementType.TYPE}∏
               @Retention □ Retention Policy. RUNTIME □
               @Documented
              @Inherited
              @EnableBinding∏{Processor.class}∏
              @Import∏{RxJavaProcessorConfiguration.class}∏
              public @interface EnableRxJavaProcessor {
          ● 🛮 App1 🖺 🖺 🖺 🖺 🖺 RxJavaProcessor 🖺 🖺 🗎 Bean 🖺 🖺
RxJavaProcessor
public interface RxJavaProcessor<I,O> {
                        Observable<O> process[]Observable<I> input[];
               }
          ●□□□□□□□□□□□□□□□RxJava□□□□□□□□□□□□□□
```

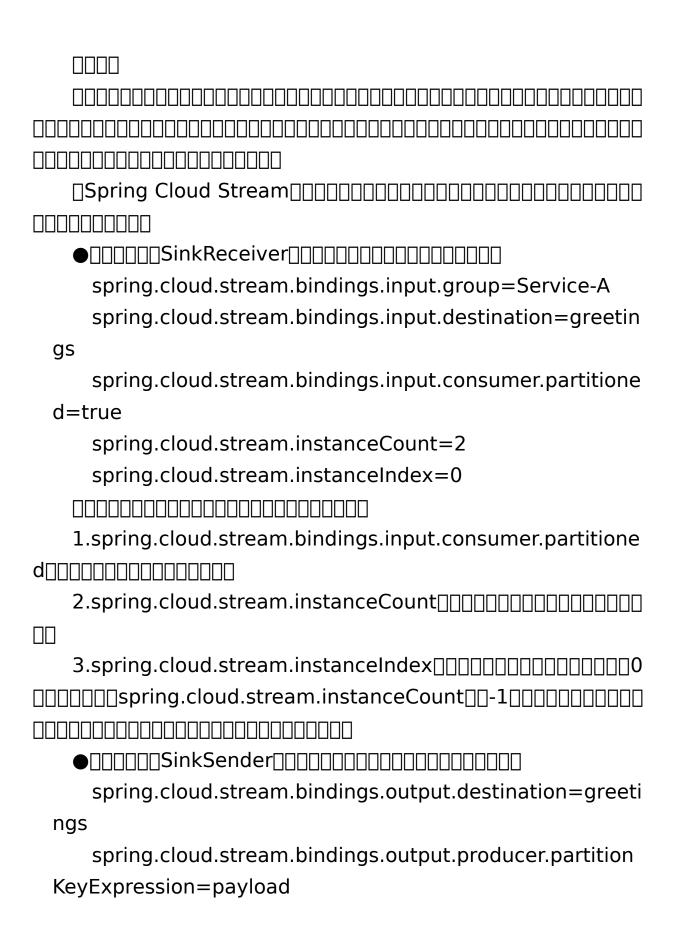
```
@EnableRxJavaProcessor
     public class App1 {
      @Bean
      public RxJavaProcessor<String,String> processor□□{
      private
                           static
                                              Logger
   logger=LoggerFactory.getLogger∏App1.class∏;
        return inputStream-> inputStream.map

☐data-> {
          logger.info∏"Received: "+data∏;
          return data:
        } □.buffer □ 5 □.map □ data-> String.valueOf □ "From
     Input Channel Return-"+data∏;
     }
   8796---[-maOhv-th7Raw-
      INFO
   1]com.didispace.HelloApplication
                                       : Received:
     From
                 Input
                               Channel
                                              Return-
 [1479390916408,1479390918409,1479390920412,14793
 90922414,
     14793909244151
      INFO
                               8796---[-maOhv-th7Raw-
   1]com.didispace.HelloApplication
                                       : Received:
     From
                 Input
                              Channel
                                              Return-
 [1479390926416,1479390928418,1479390930419.14793
 90932421,
```

1479390934423]

| | 000000000000000000000000000000000000000 | |
|---|---|-------------------|
| | | |
| | 000000000000000000000000000000000000000 | 00000000000000000 |
| 000000000000000000000000000000000000000 | | 000000000000 |
| spring.cloud.stream.bind | dings.input.group <u>□</u> □□ | |
| ●□□□□□□□□□□Sink | :Receivergreeting | s |
| | | |
| @EnableBinding[]\ | /alue={Sink.class}□ | |
| public class SinkRe | eceiver { | |
| private | static | Logger |
| logger=LoggerFacto | ry.getLogger <u></u> SinkRe | ceiver.class[]; |
| @StreamListene | er[]Sink.INPUT[] | |
| public void recei | ve]User user[]{ | |
| logger.info∏"Ro | eceived: "+user[]; | |
| } | | |
| } | | |
| ●□□□ SinkReceiver [|]greetings | s 0000000000 |
| | | |
| spring.cloud.strea | | |

| spring.cloud.stream.bind | lings.input.destination=greetin |
|---------------------------------|---|
| gs | |
| □□spring.cloud.stream.bing | dings.input.group[][][][][][][][] |
| ☐ Service-A | |
| spring.cloud.stream.bindings.ii | nput.destination[][][][][][][][][] |
| | |
| ullet |]SinkSender |
| @EnableBinding□value= | {Source.class}[] |
| public class SinkSender | { |
| private | static Logger |
| logger=LoggerFactory.getL | _ogger[SinkSender.class[]; |
| @Bean | |
| @InboundChannelAda | oter |
| □ value= | Source.OUTPUT,poller=@Poller |
| □fixedDelay="2000"□□ | |
| public MessageSource | <string> timerMessageSource</string> |
| □□{ | |
| return 🛮 🖺 -> n | ew GenericMessage<> 🛭 " |
| {\"name\":\"didi\",\"age\' | ':30}"□; |
| } | |
| } | |
| ●□□□□□□ SinkSender □□□□ |]greetings_ |
| | |
| spring.cloud.stream.bind | lings.output.destination=greeti |
| ngs | |
| | 000000000000000000000000000000000000000 |
| | |



| spring.cloud.stream.bindings.output.producer.partition |
|---|
| Count=2 |
| |
| 1. spring. cloud. stream. bindings. output. producer. partition |
| KeyExpression |
| |
| 2.spring.cloud.stream.bindings.output.producer.partition |
| Count |
| |
| |
| |
| |
| |
| Spring Cloud Stream |
| |
| Stream |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| DDDSpring Cloud Stream |
| Spring Cloud Stream 🛮 🖺 🗎 spring.cloud.stream.bindngs. |
| <channelname>.content-type</channelname> |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Stream |
| ●JSON□POJO□□□□□ |
| ●JSON□org.springframework.tuple.Tuple□□□□□□ |
| ●Object byte[] |
| byteObject |

| ●String[byte[][[][[] |
|---|
| ●Object□□□□□□Object□□□□toString□□□□ |
| JSON byteJSON |
| Object |
| |
| MIME |
| ☐ Spring Cloud Stream ☐ ☐ ☐ ☐ content-type ☐ ☐ ☐ ☐ ☐ Media |
| Type Internet Media Type |
| $application/json \verb text/plain; charset = UTF-8 \verb \verb \verb \verb HTTP \verb \verb \verb $ |
| MIMEString_byte[] |
| One MIME One of Java One of One of Java One of One of |
| java-object 🛮 🗎 🗎 🗎 🗎 🗎 🗎 application/x-javaobject; |
| type=java.util.Map |
| application/x-java-object; type=com.didispace.User \square |
| $\verb $ |
| □□application/x-spring-tuple□□□Spring□Tuple□□□ |
| Spring Cloud Stream |

| 源内容类型 | 目标内容类型 | content-type 头 | content-type | 注释 |
|-------------|-----------------|-----------------------------|---------------------|----------------------------|
| РОЈО | JSON String | ignored | application/json | |
| Tuple | JSON String | ignored | application/json | JSON is tailored for Tuple |
| РОЈО | String | ignored | text/plain, | |
| | (toString()) | | java.lang.String | |
| РОЈО | byte[] (java.io | ignored | application/x-java- | |
| | serialized) | 3537 | serialized-object | |
| JSON byte[] | РОЈО | application/json (or none) | application/x-java- | |
| 或 String | | | object | |
| byte[]或 | Serializable | application/x-java-serializ | application/x-java- | |
| String | | ed-object | object | |
| JSON byte[] | Tuple | application/json (or none) | application/x-sprin | |
| 或 String | | | g-tuple | |
| byte[] | String | any | text/plain, | will apply any Charset |
| | | | java.lang.String | specified in the |
| | | | | content-type header |
| String | byte[] | any | application/octet-s | will apply any Charset |
| | | | tream | specified in the |
| | | | | content-type header |

| $application/json \verb XML \verb $ |
|---|
| |
| Spring Cloud Stream |
| |
| @StreamListener |
| Spring Cloud Stream |
| |
| contentType |
| org.spring framework.mess aging.converter. Message Converte |
| r0000000000000000000000000000000000000 |



| _"BinderB |
|-----------|
| |
| |

| 000 SPI 00000000000000000000000000000000 |
|---|
| 0000000000000000000Binder |
| public interface Binder <t,c extends<="" td=""></t,c> |
| ConsumerProperties,P extends |
| ProducerProperties> { |
| Binding <t> bindConsumer [] String name,String</t> |
| group,T inboundBindTarget,C |
| consumerProperties[]; |
| Binding <t> bindProducer [] String name,T</t> |
| outboundBindTarget,P |
| producerProperties[]; |
| } |
| |
| ●□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| |
| ●□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| |
| |
| BinderBinder |

| ullet |
|---|
| |
| ●□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| |
| Binder |
| ●□□□□Binder□□□□ |
| ●□□Spring□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ |
| ●□□□□□□□□classpath□□META-INF/spring.binders□□□□□□□ |
| $\verb $ |
| |
| rabbit:\ |
| org.spring framework.cloud.stream.binder.rabbit.config. |
| RabbitServiceAutoConfigur ation |
| |
| |
| Spring Cloud Stream |
| |
| |
| |
| DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| |
| |
| |
| |
| |
| |

```
<artifactId>spring-cloud-stream-binder-
  rabbit</artifactId>
   </dependency>
  \square\square\square\square\squareKafka\square\square\square\square\square
   <dependency>
     <groupId>org.springframework.cloud</groupId>
     <artifactId>spring-cloud-stream-binder-
  kafka</artifactId>
   </dependency>
                 □□□□□□classpath□□□□□□□□□□□□Spring Cloud Stream□□□□□□
spring.cloud.stream.defaultBinder=rabbit
  spring.cloud.stream.bindings.input.binder=kafka
  OMETA-INF/spring.binders
_____rabbit_kafka________
   rabbit:\
   org.springframework.cloud.stream.binder.rabbit.config.
 RabbitServiceAutoConfigur ation
   kafka:\
```

| KafkaBinderConfiguration |
|--|
| |
| RabbitMQ |
| spring.cloud.stream.bindings.input.binder=rabbit1 |
| spring.cloud.stream.bindings.output.binder=rabbit2 |
| spring.cloud.stream.binders.rabbit 1.type = rabbit |
| spring. cloud. stream. binders. rabbit 1. environment. sprin |
| g.rabbitmq.host=192.168.0.101 |
| spring.cloud.stream.binders.rabbit 1. environment.sprin |
| g.rabbitmq.port=5672 |
| spring.cloud.stream.binders.rabbit1.environment.sprin |
| g.rabbitmq.username=springcloud |
| spring.cloud.stream.binders.rabbit1.environment.sprin |
| g.rabbitmq.password=123456 |
| spring.cloud.stream.binders.rabbit2.type=rabbit |
| spring.cloud.stream.binders.rabbit2.environment.sprin |
| g.rabbitmq.host=192.168.0.102 |
| spring.cloud.stream.binders.rabbit2.environment.sprin |
| g.rabbitmq.port=5672 |
| spring.cloud.stream.binders.rabbit2.environment.sprin |
| g.rabbitmq.username=springcloud |
| spring.cloud.stream.binders.rabbit2.environment.sprin |
| g.rabbitmq.password=123456 |
| |
| rabbit1outputrabbit2 |
| |

org.spring framework.cloud.stream.binder.kafka.config.

| spring.cloud.stream.binders. <configurationname>\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ </configurationname> |
|--|
| <pre>spring.cloud.stream.binders. <configurationname>.type </configurationname></pre> |
| <configurationname>.environment</configurationname> |
| |
| • spring.cloud.stream.binders. |
| <configurationname>.inheritEnvir onment Output Description Output Description Output Description Output Description Output Description De</configurationname> |
| |
| • spring.cloud.stream.binders. |
| <configurationname>.defaultCandi date</configurationname> |
| |
| true_ |
| |
| |
| |
| |
| RabbitMQ\Kafka\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ |
| RabbitMQ\Kafka\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ |
| RabbitMQ\Kafka\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ |
| RabbitMQ Kafka CONDON RabbitMQ Rafka CONDON |
| |
| RabbitMQ Kafka |



Spring Cloud Stream



| 参数名 | 说明 | 默认值 |
|-------------------------|--------------------------------------|-------|
| instanceCount | 应用程序部署的实例数量。当使用 Kafka 的时候需要设置 | 1 |
| | 分区 | |
| instanceIndex | 应用程序实例的索引,该值从0开始,最大值设置为-1。 | |
| | 当使用分区和 Kafka 的时候使用 | |
| dynamicDestinations | 动态绑定的目标列表,该列表默认为空,当设置了具体列 | 空 |
| | 表之后,只有列表中的目标才能被发现 | |
| defaultBinder | 默认绑定器配置,在应用程序中有多个绑定器时使用 | 空 |
| overrideCloudConnectors | 该属性只适用于激活 cloud 配置并且提供了 Spring Cloud | false |
| | Connectors 的应用。当使用默认属性 false 时,绑定器会 | |
| | 自动检测合适的服务来绑定(比如,在 Cloud Foundry 中 | |
| | 绑定的 RabbitMQ 服务)。当设置为 true 时,绑定器将忽 | |
| | 略绑定的服务,而是依赖应用程序中的设置属性来进行绑 | |
| | 定和连接 | |



| spring.cloud.stream.bindings. <channelname>.<property>=</property></channelname> |
|--|
| <value>000000000000000000000000000000000000</value> |
| |
| |
| |
| |
| <pre><channelname></channelname></pre> |
| |
| |
| $spring.cloud.stream.bindings. < channel Name > . \verb $ |
| |

| 参数名 | 说明 | 默认值 | | | | | |
|-------------|--|------|--|--|--|--|--|
| destination | 该参数用来配置消息通道绑定在消息中间件中的目标名称,比如 RabbitMQ | | | | | | |
| | 的 Exchange 或 Kafka 的 Topic。如果配置的绑定通道是一个消费者(输 | | | | | | |
| | 通道),那么它可以绑定多个目标,这些目标名称通过逗号来分隔。如果 | | | | | | |
| | 没有设置该属性,将使用通道名 | | | | | | |
| group | 该参数用来设置绑定通道的消费组,该参数主要作用于输入通道,以保证 | null | | | | | |
| | 同一消息组中的消息只会有一个消费示例接收和处理 | | | | | | |
| contentType | 该参数用来设置绑定通道的消息类型 | null | | | | | |
| binder | 当存在多个绑定器时使用该参数来指定当前通道使用哪个具体的绑定器 | null | | | | | |

_____spring.cloud.stream.bindings.

<channelName>.consumer. $\Box\Box\Box\Box\Box\Box\Box$

| 参数名 | 说明 | 默认值 |
|------------------------|--|-----------------|
| concurrency | 输入通道消费者的并发数 | 1 |
| partitioned | 来自消息生产者的数据是否采用了分区 | false |
| headerMode | 当设置为 raw 的时候将禁用对消息头的解析。
该属性只有在使用不支持消息头功能的中间件
时有效,因为 Spring Cloud Stream 默认会解析
嵌入的头部信息 | embeddedHeaders |
| maxAttempts | 对输入通道消息处理的最大重试次数 | 3 |
| backOffInitialInterval | 重试消息处理的初始间隔时间 | 1000 |
| backOffMaxInterval | 重试消息处理的最大间隔时间 | 10000 |
| backOffMultiplier | 重试消息处理时间间隔的递增乘数 | 2.0 |

| | _ | _ | _ | _ |
|--|---|----|----|------|
| | | 11 | " | II I |
| | | 11 | 11 | II I |
| | | 11 | " | II I |
| | | | | |

<channelName>.producer.

| 参数名 | 说明 | 默认值 |
|------------------------|---------------------------------|------|
| partitionKeyExpression | 该参数用来配置输出通道数据分区键的 SpEL | null |
| | 表达式。当设置该属性之后,将对当前绑定通 | |
| | 道的输出数据进行分区处理。同时, | |
| | partitionCount参数必须大于1才能生效。 | |
| | 该参数与 partitionKeyExtractorClass | |
| | 参数互斥,不能同时设置 | |

| 参数名 | 说明 | 默认值 |
|-----------------------------|--|-----------------|
| partitionKeyExtractorClass | 该参数用来配置分区键提取策略接口
PartitionKeyExtractorStrategy的实现。当设置该属性之后,将对当前绑定通道的输出数据进行分区处理。同时,
partitionCount参数必须大于1才能生效。该参数与 partitionKeyExpression
参数互斥,不能同时设置 | null |
| partitionSelectorClass | 该参数用来指定分区选择器接口PartitionSelectorStrategy的实现。它与partitionSelectorExpression参数互斥,不能同时设置。如果两者都不设置,那么分区选择计算规则为 hashCode(key)% partitionCount,这里的 key 根据partitionKeyExpression或partition-KeyExtractorClass的配置计算得到 | null |
| partitionSelectorExpression | 该参数用来配置自定义分区选择器的 SpEL 表达式。它与 partitionSelectorClass 参数互斥,不能同时设置。如果两者都不设置,那么分区选择计算规则为 hashCode (key) % partitionCount ,这里的 key 根据 partitionKeyExpression或 partitionKeyExtractorClass的配置计算得到 | null |
| partitionCount | 当分区功能开启时,使用该参数来配置消息数据的分区数。如果消息生产者已经配置了分区键的生成策略,那么它的值必须大于1 | 1 |
| headerMode | 当设置为 raw 的时候将在输出时禁用嵌入式
头部信息。该属性只有在使用不支持消息头功
能的中间件时有效,因为 Spring Cloud Stream
会默认地为其嵌入头部信息 | embeddedHeaders |

| Spring Cloud Stream RabbitMQ Kafka |
|---|
| |
| |

 $\textbf{RabbitMQ} \underline{\square} \underline{\square}$

| | RabbitMQ[[| | |
|-----------------|---------------------------------------|---|----------------|
| | | | |
| | | | |
| | □□ RabbitM | Q | ctorv⊓ |
| | |]Spring Bootspring.rabbi | |
| | | | |
| | | | |
| ∏Sp | oring Boot∏∏ | | tarter- |
| am | qp□□□Rabbit | :Properties | |
| | □Spring Clo | ud Stream RabbitMQ 000000000000000000000000000000000000 | |
| Πs | pring.cloud. | stream.rabbit.binder. ☐ ☐ ☐ ☐ ☐ ☐ ☐ | ппп |
| _ | | ework.cloud.stream.binder.rabbit.config.I | a a a
Rahhi |
| org | .spinighann | Work.cloud.3tream.binder.rabbit.comig.i | Kabbi |
| +Dir | odorConfigur | sation Dranartics PPPPP | |
| tBir | nderConfigur | rationProperties 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 🗆 | |
| | nderConfigur
_{数名} | rationProperties[][][][][]
^{说明} | 默认值 |
| 参 | _ | | 默认值 |
| 参 | 数名 | 说明
该参数用来配置 RabbitMQ 管理插件的 URL, 当需要配置多个时用
逗号分隔。该参数只有在 nodes 参数包含多个时使用, 并且这里配 | 默认值 |
| 参 | 数 名
dminAddresses | 说明
该参数用来配置 RabbitMQ 管理插件的 URL,当需要配置多个时用
逗号分隔。该参数只有在 nodes 参数包含多个时使用,并且这里配
置的内容必须在 spring.rabbitmq.addresses 中存在 | 默认值 |
| 参 | 数名 | 说明
该参数用来配置 RabbitMQ 管理插件的 URL,当需要配置多个时用
逗号分隔。该参数只有在 nodes 参数包含多个时使用,并且这里配
置的内容必须在 spring.rabbitmq.addresses 中存在
该参数用来配置 RabbitMQ 的节点名称,当需要配置多个时用逗号分 | 默认值 |
| 参 | 数 名
dminAddresses | 说明
该参数用来配置 RabbitMQ 管理插件的 URL,当需要配置多个时用
逗号分隔。该参数只有在 nodes 参数包含多个时使用,并且这里配
置的内容必须在 spring.rabbitmq.addresses 中存在
该参数用来配置 RabbitMQ 的节点名称,当需要配置多个时用逗号分
隔。在配置多个的情况下,可以用来定位队列所在的服务器地址。这 | 默认值 |
| 参
acc | 数 名
dminAddresses | 说明
该参数用来配置 RabbitMQ 管理插件的 URL,当需要配置多个时用
逗号分隔。该参数只有在 nodes 参数包含多个时使用,并且这里配
置的内容必须在 spring.rabbitmq.addresses 中存在
该参数用来配置 RabbitMQ 的节点名称,当需要配置多个时用逗号分 | 默认值 |
| 参
acc | 数名
dminAddresses
odes | 说明 该参数用来配置 RabbitMQ 管理插件的 URL,当需要配置多个时用逗号分隔。该参数只有在 nodes 参数包含多个时使用,并且这里配置的内容必须在 spring.rabbitmq.addresses 中存在该参数用来配置 RabbitMQ 的节点名称,当需要配置多个时用逗号分隔。在配置多个的情况下,可以用来定位队列所在的服务器地址。这里配置的内容必须在 spring.rabbitmq.addresses 中存在 | |
| 参
acc | 数名
dminAddresses
odes | 说明 该参数用来配置 RabbitMQ 管理插件的 URL,当需要配置多个时用逗号分隔。该参数只有在 nodes 参数包含多个时使用,并且这里配置的内容必须在 spring.rabbitmq.addresses 中存在该参数用来配置 RabbitMQ 的节点名称,当需要配置多个时用逗号分隔。在配置多个的情况下,可以用来定位队列所在的服务器地址。这里配置的内容必须在 spring.rabbitmq.addresses 中存在绑定通道的压缩级别,它的具体可选值及含义可见 java.util. | |
| 参
acc | 数名 dminAddresses odes ompressionLevel | 说明 该参数用来配置 RabbitMQ 管理插件的 URL,当需要配置多个时用逗号分隔。该参数只有在 nodes 参数包含多个时使用,并且这里配置的内容必须在 spring.rabbitmq.addresses 中存在该参数用来配置 RabbitMQ 的节点名称,当需要配置多个时用逗号分隔。在配置多个的情况下,可以用来定位队列所在的服务器地址。这里配置的内容必须在 spring.rabbitmq.addresses 中存在绑定通道的压缩级别,它的具体可选值及含义可见 java.util. | 1 |
| 参
acc | 数名 dminAddresses odes ompressionLevel | 说明 该参数用来配置 RabbitMQ 管理插件的 URL,当需要配置多个时用 逗号分隔。该参数只有在 nodes 参数包含多个时使用,并且这里配置的内容必须在 spring.rabbitmq.addresses 中存在 该参数用来配置 RabbitMQ 的节点名称,当需要配置多个时用逗号分隔。在配置多个的情况下,可以用来定位队列所在的服务器地址。这里配置的内容必须在 spring.rabbitmq.addresses 中存在 绑定通道的压缩级别,它的具体可选值及含义可见 java.util.zip.Deflater中的定义 | 1 |

| 参数名 | 说明 | 默认值 |
|-----------------|---------------|------|
| acknowledgeMode | 用来设置消息的确认模式,可 | AUTO |
| | 选配置包含: NONE、 | |
| | MANUAL, AUTO | |

| 参数名 | 说明 | 默认值 |
|-----------------------|----------------------------|--------------------------------|
| autoBindDlq | 用来设置是否自动声明 DLQ | false |
| | (Dead-Letter-Queue),并绑定到 | |
| | DLX (Dead-Letter-Exchange) | |
| | 上 | |
| durableSubscription | 用来设置订阅是否被持久化, | true |
| | 该参数仅在 group 被设置的 | |
| | 时候有效 | |
| maxConcurrency | 用来设置消费者的最大并发数 | 1 |
| prefetch | 用来设置预取数量,它表示在 | 1 |
| 60 | 一次会话中从消息中间件中 | |
| | 获取的消息数量,该值越大消 | |
| | 息处理越快,但是会导致非顺 | |
| | 序处理的风险 | |
| prefix | 用来设置统一的目标和队列 | |
| 8 | 名称前缀 | |
| recoveryInterval | 用来设置恢复连接的尝试时 | 5000 |
| | 间间隔,以毫秒为单位 | |
| requeueRejected | 用来设置消息传递失败时重传 | true |
| requestHeaderPatterns | 用来设置需要被传递的请求 | [STANDARD_REQUEST_HEADERS,'*'] |
| 50.00 | 头信息 | |
| replyHeaderPatterns | 用来设置需要被传递的响应 | [STANDARD_REPLY_HEADERS,'*'] |
| 76 7 TS | 头信息 | 20611 5395 |
| republishToDlq | 默认情况下,消息在重试也失 | |
| | 败之后会被拒绝。如果 DLQ | |
| | 被配置的时候, RabbitMQ 会 | |
| | 将失败的消息路由到 DLQ | |
| | 中。如果该参数设置为 true, | |
| | 总线会将失败的消息附加一 | |
| | 些头信息(包括异常消息、引 | |
| | 起失败的跟踪堆栈)之后重新 | |
| | 发布到 DLQ 中 | |

| 参数名 | 说明 | 默认值 |
|------------|-----------------------|-------|
| transacted | 用来设置是否启用 channel- | false |
| | transacted,即是否在消息 | |
| | 中使用事务 | |
| txSize | 用来设置 transaction-size | 1 |
| | 的数量,当 acknowledge- | |
| | Mode 被设置为 AUTO 时,容 | |
| | 器会在处理 txSize 数目消 | |
| | 息之后才开始应答 | |

| 🛮 🗎 🗎 🗎 🗎 🗎 RabbitMQ | | | | | | | |
|---------------------------------------|------|--|--|--|--|--|--|
| spring.cloud.stream.rabbit.bindi | ngs. | | | | | | |
| <channelname>.producer.</channelname> | | | | | | | |

| 参数名 | 说明 | 默认值 |
|-----------------------|----------------------------|--------------------------------|
| autoBindDlq | 用来设置是否自动声明 DLQ | false |
| | (Dead-Letter-Queue),并绑定到 | |
| | DLX (Dead-Letter-Exchange) | |
| | 上 | |
| batchingEnabled | 是否启用消息批处理 | false |
| batchSize | 当批处理开启时,用来设置 | 100 |
| | 缓存的批处理消息数量 | |
| batchBufferLimit | 批处理缓存限制 | 10000 |
| batchTimeout | 批处理超时时间 | 5000 |
| compress | 消息发送时是否启用压缩 | false |
| deliveryMode | 消息发送模式 | PERSISTENT |
| prefix | 用来设置统一的目标前缀 | |
| requestHeaderPatterns | 用来设置需要被传递的请求 | [STANDARD_REQUEST_HEADERS,'*'] |
| | 头信息 | |
| replyHeaderPatterns | 用来设置需要被传递的响应 | [STANDARD_REPLY_HEADERS,'*'] |
| | 头信息 | |

Kafka∏

| Kafka 🗆 🗆 🗆 🗆 🗆 Rabbit M | 1Q000000000000000000000000000000000000 |
|--------------------------|--|
| □□□□□□ RabbitMQ 🛮 Kafka | |
| | |

Spring Cloud Stream [] Kafka [] [] [] Spring.cloud.stream.kafka.binder.

| 参数名 | 说明 | 默认值 |
|------------------------|--|-----------|
| brokers | Kafka 绑定器连接的消息中间件列表。需要配置多个时用 逗号分隔,每个地址可以是单独的 host,也可以是 host:port的形式 | localhost |
| defaultBrokerPort | 用来设置默认的消息中间件端口号。当 brokers 中的配置地址没有包含端口信息时,将使用该参数配置的默认端口进行连接 | 9092 |
| zkNodes | Kafka 绑定器使用的 ZooKeeper 节点列表。需要配置多个时用逗号分隔,每个地址可以是单独的 host, 也可以是host:port 的形式 | localhost |
| defaultZkPort | 用来设置默认的 ZooKeeper 端口号。当 zkNodes 中的配置地址没有包含端口信息时,将使用该参数配置的默认端口进行连接 | 2181 |
| headers | 用来设置会被传输的自定义头信息 | |
| offsetUpdateTimeWindow | 用来设置 offset 的更新频率,以毫秒为单位,如果设置为 0 则忽略 | 10000 |
| offsetUpdateCount | 用来设置 offset 以次数表示的更新频率,如果为 0 则忽略,该参数与 offsetUpdateTimeWindow 互斥 | 0 |
| requiredAcks | 用来设置确认消息的数量 | 1 |
| minPartitionCount | 该参数仅在设置了 autoCreateTopics 和 autoAdd-Partitions 时生效,用来设置该绑定器所使用主题的全局分区最小数量。如果当生产者的 partitionCount 参数或 instanceCount * concurrency 设置大于该参数配置时,该参数值将被覆盖 | 1 |
| replicationFactor | 当 autoCreateTopics 参数为 true 时候,用来配置自
动创建主题的副本数量 | 1 |
| autoCreateTopics | 该参数默认为 true, 绑定器会自动地创建新主题。如果设置为 false, 那么绑定器将使用已经配置的主题, 但是在这种情况下, 如果需要使用的主题不存在, 绑定器会启动失败 | true |

| 参数名 | 说明 | 默认值 |
|-------------------|---|---------|
| autoAddPartitions | 该参数默认为 false, 绑定器会根据已经配置的主题分区来实现, 如果目标主题的分区数小于预期值, 那么绑定器会启动失败。如果该参数设置为 true, 绑定器将在需要的时候自动创建新的分区 | false |
| socketBufferSize | 该参数用来设置 Kafka 的 Socket 缓存大小 | 2097152 |

| | Kafka | | | | | | | |
|-----------------------------------|----------|-------|---|--|--|--|--|--|
| spring.cloud.stream.kaf | ka.bindi | ings. | | | | | | |
| <channelname>.consu</channelname> | mer.[][[| |] | | | | | |

| 参数名 | 说明 | 默认值 |
|-------------------|--|-------|
| autoCommitOffset | 用来设置是否在处理消息时自动提交 offset。如果设置为 false, | true |
| | 在消息头中会加入 ACK 头信息以实现延迟确认 | |
| autoCommitOnError | 该参数只有在 autoCommitOffset 设置为 true 时才有效。当 | |
| | 设置为 false 的时候,引起错误的消息不会自动提交 offset, | |
| | 仅提交成功消息的 offset。如果设置为 true, 不论消息是否成 | |
| | 功,都会自动提交。当不设置该值时,它实际上具有与enableDlq | |
| | 相同的配置值 | |
| recoveryInterval | 尝试恢复连接的时间间隔,以毫秒为单位 | 5000 |
| resetOffsets | 是否使用提供的 startOffset 值来重置消费者的 offset 值 | false |
| startOffset | 用来设置新建组的起始 offset, 该值也会在 resetOffsets | null |
| | 开始时被使用 | |
| enableDlq | 该参数设置为 true 时,将为消费者启用 DLQ 行为,引起错误的 | false |
| | 消息将被发送到名为 error. <destination>.<group>的主</group></destination> | |
| | 题中去 | |

| | Kafka | | | | | | | | |
|------------------------------------|-----------|-----|----|--|--|--|--|--|--|
| spring.cloud.stream.kafl | ka.bindi | ing | S. | | | | | | |
| <channelname>.produc</channelname> | cer.[][[[| | | | | | | | |

| 参数名 | 说明 | 默认值 |
|------------|---|-------|
| bufferSize | Kafka 批量发送前的缓存数据上限,以字节为单位 | 16384 |
| sync | 该参数用来设置 Kafka 消息生产者的发送模式,默认为 false,即采用 async 配置,允许批量发送数据。当设置为 true 时,将采用 sync 配置,消息将不会被批量发送,而是一条一条地发送 | false |

| 参数名 | 说明 | 默认值 |
|--------------|------------------------------------|-----|
| batchTimeout | 消息生产者批量发送时,为了积累更多发送数据而设置的等待时间。通常 | 0 |
| | 情况下,生产者基本不会等待,而是直接发送所有在前一批次发送时积累 | |
| | 的消息数据。当我们设置一个非 0 值时,可以以延迟为代价来增加系统的 | |
| | 吞吐量 | |

□11□ □□□□□□Spring Cloud Sleuth



_____Sleuth



```
20000000000000000
  ■□□□□□□Spring Boot□□□□pom.xml□□□□□□□□
    <parent>
       <groupId>org.springframework.boot</groupId>
       <artifactId>spring-boot-starter-parent</artifactId>
       <version>1.3.7.RELEASE
       <relativePath/>
    </parent>
    <dependency>
       <groupId>org.springframework.boot</groupId>
       <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
       <groupId>org.springframework.cloud
       <artifactId>spring-cloud-starter-
    eureka</artifactId>
    </dependency>
    <dependency>
       <groupId>org.springframework.cloud</groupId>
       <artifactId>spring-cloud-starter-
    ribbon</artifactId>
    </dependency>
    <dependencyManagement>
       <dependencies>
```

```
<dependency>
             <groupId>org.springframework.cloud
          pld>
             <artifactId>spring-cloud-
          dependencies</artifactId>
             <version>Brixton.SR5</version>
             <type>pom</type>
             <scope>import</scope>
          </dependency>
        </dependencies>
     </dependencyManagement>
   @RestController
     @EnableDiscoveryClient
     @SpringBootApplication
     public class TraceApplication {
      private
                             logger=Logger.getLogger
               final
                     Logger
   □getClass□□□;
      @Bean
      @LoadBalanced
      RestTemplate restTemplate □ {
        return new RestTemplate∏;
                                      value="/trace-
      @RequestMapping
                              П
   1",method=RequestMethod.GET□
      public String trace [ [ ] {
```

```
return restTemplate [ ] .getForEntity [] "http://trace-
    2/trace-2",String.class∏.getBody∏∏;
      }
      public static void main

String[]args

{
        SpringApplication.run
☐TraceApplication.class,args☐;
      }
     }
               application.properties
                                     П
                                           П
eureka.client.serviceUrl.defaultZone
spring.application.name=trace-1
    server.port=9101
    eureka.client.serviceUrl.defaultZone=http://localhost:1
 111/eureka/
   ■□□□□□□Spring Boot□□□pom.xml□□□□□trace-1□□□
   ■□□□□□□□□□/trace-2□□□□□□□□□□
     @RestController
    @EnableDiscoveryClient
    @SpringBootApplication
    public class TraceApplication {
      private
              final
                     Logger
                             logger=Logger.getLogger
   ∏getClass∏∏;
      @RequestMapping
                                      value="/trace-
                             П
   2",method=RequestMethod.GET[]
```

```
public String trace [[] {
       logger.info[]"===<call trace-2>==="[];
       return "Trace";
      }
      public static void main

☐String[]args
☐{
       SpringApplication.run
☐TraceApplication.class,args☐;
      }
    }
              application.properties
                                    П
                                         П
eureka.client.serviceUrl.defaultZone
spring.application.name=trace-2
    server.port=9102
    eureka.client.serviceUrl.defaultZone=http://localhost:1
 111/eureka/
   _____ postman __ curl ____ trace-1 _____
http://localhost:9101/trace-1
--trace-1
                             25272---[nio-9101-exec-
      INFO
   2]ication$$EnhancerBySpringCGLIB$$36e12c68:
    ===<call trace-1>===
      --trace-2
                              7136---[nio-9102-exec-
      INFO
   1]ication$$EnhancerBySpringCGLIB$$52a02f0b
                                           :===
   <call
```

```
One of the state o
∏∏∏ trace-1∏ trace-2∏ pom.xml ∏∏∏∏∏∏∏spring-cloud-starter-
sleuth
                   <dependency>
                         <groupId>org.springframework.cloud</groupId>
                         <artifactId>spring-cloud-starter-sleuth</artifactId>
                   </dependency>
            ____Sleuth
                         --trace-1
                        INFO[trace-
            1,f410ab57afd5c145,a9f2118fa2019684,false]25028---
            [nio-9101-exec-1]
                  ication$$EnhancerBySpringCGLIB$$d8228493
       <call trace-1>===
                         --trace-2
                         INFO[trace-
            2,f410ab57afd5c145,e9a377dc2268bc29,false]23112---
            [nio-9102-exec-1]
                  ication$$EnhancerBySpringCGLIB$$e6cb4078
                                                                                                                                                                              :===
       <call trace-2>===
```

[trace-
$1,\!f410ab57afd5c145,\!a9f2118fa2019684,\!false] \verb $
ullet $lacksquare$ $lack$
spring.application.name
●□□□□□f410ab57afd5c145,Spring Cloud Sleuth□□□□□ID□□
Trace IDSpan ID
$ullet$ \bigcap \big
IDSpan IDHTTP
●□□□□□false□□□□□□□□□□□□□Zipkin□□□□□□□□
00000000000000000000000000000000000000

```
_____spring-cloud-
starter-sleuth
● □ □ □ □ Rabbit MQ □ Kafka □ □ □ □ □ □ Spring Cloud Stream □ □ □ □
● □ RestTemplate □ □ □ □ □
 ●X-B3-ParentSpanId□□□□□□□□□□□□□□□□□□□Root Span□□
@RequestMapping
              П
                  value="/trace-
2",method=RequestMethod.GET□
  public String trace[HttpServletRequest request[]{
   logger.info □ "===<call trace-2,TraceId={},SpanId=
 {}>===",
    request.getHeader
                      "X-B3-
  TraceId"∏,request.getHeader∏"X-B3-SpanId"∏∏;
   return "Trace";
  }
```

```
--trace-1
    INFO[trace-
  1,a6e9175ffd5d2c88,8524f519b8a9e399,true]10532---
  [nio-9101-exec-2]
   icationEnhancerBySpringCGLIB27aa9624 :===<call
 trace-1>===
    --trace-2
    INFO[trace-
  2,a6e9175ffd5d2c88,ce60dcf1e2ed918f,true]1208---[nio-
  9102-exec-3]icationEnhancerBySpringCGLIBa7d84797
  :===<call trace-2, TraceId=a6e9175ffd5d2c88,
   SpanId=be4949ec115e554e>===
  logging.level.org.springframework.web.servlet.Dispatc
 herServlet=DEBUG
```

```
-- trace-1
    2016-11-27 09:26:52.663 DEBUG [trace-1,a6e9175ffd5d2c88,a6e9175ffd5d2c88,true]
10532 --- [nio-9101-exec-2] o.s.web.servlet.DispatcherServlet
DispatcherServlet with name 'dispatcherServlet' processing GET request for [/trace-1]
    2016-11-27 09:26:52.666 DEBUG [trace-1,a6e9175ffd5d2c88,a6e9175ffd5d2c88,true]
10532 --- [nio-9101-exec-2] o.s.web.servlet.DispatcherServlet : Last-Modified
value for [/trace-1] is: -1
    2016-11-27 09:26:52.685 DEBUG [trace-1,a6e9175ffd5d2c88,8524f519b8a9e399,true]
10532 --- [nio-9101-exec-2] o.s.web.servlet.DispatcherServlet : Null ModelAndView
returned to DispatcherServlet with name 'dispatcherServlet': assuming HandlerAdapter
completed request handling
    2016-11-27 09:26:52.685 DEBUG [trace-1,a6e9175ffd5d2c88,a6e9175ffd5d2c88,true]
10532 --- [nio-9101-exec-2] o.s.web.servlet.DispatcherServlet : Successfully
completed request
    -- trace-2
    2016-11-27 09:26:52.673 DEBUG [trace-2,a6e9175ffd5d2c88,be4949ec115e554e,true]
1208 --- [nio-9102-exec-3] o.s.web.servlet.DispatcherServlet : DispatcherServlet
with name 'dispatcherServlet' processing GET request for [/trace-2]
    2016-11-27 09:26:52.679 DEBUG [trace-2,a6e9175ffd5d2c88,be4949ec115e554e,true]
1208 --- [nio-9102-exec-3] o.s.web.servlet.DispatcherServlet : Last-Modified
value for [/trace-2] is: -1
    2016-11-27 09:26:52.682 DEBUG [trace-2,a6e9175ffd5d2c88,ce60dcf1e2ed918f,true]
1208 --- [nio-9102-exec-3] o.s.web.servlet.DispatcherServlet : Null ModelAndView
returned to DispatcherServlet with name 'dispatcherServlet': assuming HandlerAdapter
completed request handling
    2016-11-27 09:26:52.683 DEBUG [trace-2,a6e9175ffd5d2c88,be4949ec115e554e,true]
1208 --- [nio-9102-exec-3] o.s.web.servlet.DispatcherServlet : Successfully
completed request
```



```
Sleuth Complete Compl
                    public interface Sampler {
                          /**
                               * @return true if the span is not null and should
                    be exported to the tracing
               system
                         boolean isSampled

Span span

;
                    }
          \Pi\Pi\Pi\Pi\Pi\Pi\Pi\Pi\Pi\Pi\Pi
          spring.sleuth.sampler.percentage=0.1
          \verb||AlwaysSampler||Bean|||||||isSampled||||||||true|||||||
@Bean
               public AlwaysSampler defaultSampler□□{
                    return new AlwaysSampler∏;
          _____Tag____
               public class TagSampler implements Sampler {
```

```
private String tag;
public TagSampler[String tag[]{
    this.tag=tag;
}
@Override
public boolean isSampled[Span span[]{
    return span.tags[[].get[]tag[]!=null;
}
}
```

[[Logstash[]]

$Sleuth \verb spring-cloud-starter-sleuth $
Trace ID
ELKElasticSearch_Logstash_Kibana
●ElasticSearch [][][][][][][][][][][][][][][][][][][]
000000000RESTful
●Logstash
●Kibana □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

```
Spring Boot
● pom.xml | pom
                          <dependency>
                                           <groupId>net.logstash.logback/groupId>
                                           <artifactId>logstash-logback-encoder</artifactId>
                                           <version>4.6</version>
                          </dependency>
                 ● 🛮 🗎 /resource 🖺 🖺 🖺 🗎 bootstrap.properties 🖺 🖺 🖺 🖺
spring.xml \( \price \price \) application.properties \( \price \
logbackspring.xml | | | | | | spring.application.name | | | | | | | |
\bullet \square \square / resource \square \square \square \square logback \square \square \square logback-spring.xml \square \square \square
<? xml version="1.0" encoding="UTF-8"? >
                                  <configuration>
                                           <include
                         resource="org/springframework/boot/logging/logback/
                          defaults.xml"/>
                                                                                                                                                                                             scope="context"
                                          <springProperty</pre>
                         name="springAppName" source="spring.
                          application.name"/>
```

```
<!--00000000000-->
                                name="LOG FILE"
   property
value="${BUILD FOLDER:-
build}/${springAppName}"/>
 <!--
   property name="CONSOLE LOG PATTERN"
        value="%clr
                          П
                                 %d{yyyy-MM-dd
     HH:mm:ss.SSS}
                                {faint}
                         П
                                            %clr
         ${LOG LEVEL PATTERN:-%5p}
                                       П
                                            %clr
                    [${springAppName:-},%X{X-B3-
     Traceld:-},%X{X-B3-SpanId:-},%X{X-Span-Expo
     rt:-}]\[{yellow} %clr\[${PID:-}\[{magenta} %clr\[-
     -- \sqcap {faint} %clr \sqcap [%15.15t] \sqcap {faint} %c lr
     □ %-40.40logger{39} □ {cyan} %clr □ : □ {faint}
     %m%n${LOG EXCEPTION CONVERSION WORD:-
     %wE x"/>
   <appender
                                 name="console"
class="ch.gos.logback.core.ConsoleAppender">
     <filter
 class="ch.gos.logback.classic.filter.ThresholdFilter">
       <level>INFO</level>
     </filter>
     <encoder>
       <pattern>${CONSOLE LOG PATTERN}
   </pattern>
       <charset>utf8</charset>
```

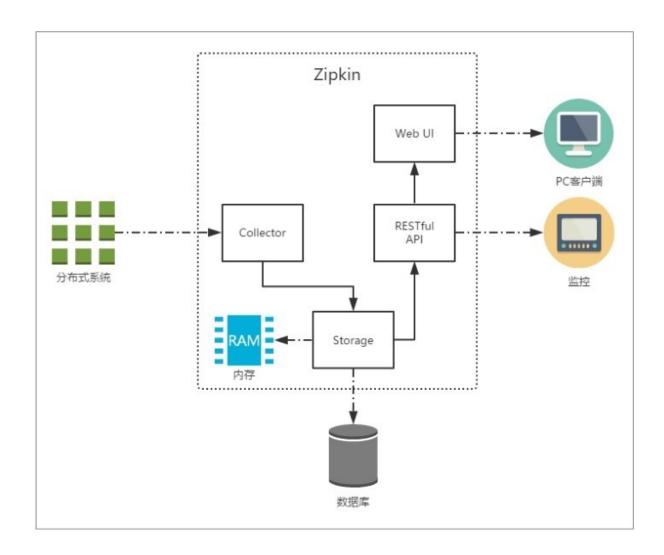
```
</encoder>
   </appender>
   <appender
                                  name="logstash"
class="ch.qos.logback.core.rolling.RollingFileAppender"
>
     <file>${LOG FILE}.json</file>
     <rollingPolicy</pre>
 class="ch.gos.logback.core.rolling.TimeBasedRolling
 Policy">
       <fileNamePattern>${LOG FILE}.json.%d{yyyy
   -MM-dd}.gz</fileNamePattern>
       <maxHistory>7</maxHistory>
     </rollingPolicy>
     <encoder
 class="net.logstash.logback.encoder.LoggingEventC
 omposite|sonEncoder">
       cproviders>
         <timestamp>
          <timeZone>UTC</timeZone>
         </timestamp>
         <pattern>
          <pattern>{
            "severity": "%level",
            "service": "${springAppName:-}",
            "trace": "%X{X-B3-TraceId:-}",
            "span": "%X{X-B3-SpanId:-}",
```

```
"exportable": "%X{X-Span-Export:-}",
            "pid": "${PID:-}",
            "thread": "%thread",
            "class": "%logger{40}",
            "rest": "%message"}
           </pattern>
         </pattern>
        </providers>
       </encoder>
      </appender>
      <root level="INFO">
       <appender-ref ref="console"/>
       <appender-ref ref="logstash"/>
      </root>
  {"@timestamp":"2016-12-
 04T06:57:58.970+00:00", "severity": "INFO", "service": "trac
 e-
 1","trace":"589ee5f7b860132f","span":"a9e891273affb7fc"
 ,"exportable":"false","pid":"19756","thread":"http-nio-
 9101-exec-1", "class": "c.d. Trace Application $$ Enhancer By
```

```
SpringCGLIB$$a9604da6","rest":"===<call
                                               trace-
 1>==="}
     {"@timestamp":"2016-12-
 04T06:57:59.061+00:00", "severity": "INFO", "service": "trac
 e-
 1","trace":"589ee5f7b860132f","span":"2df8511ddf3d79a2
 ","exportable":"false","pid":"19756","thread":"http-nio-
 9101-exec-1", "class": "o.s.c.a. Annotation Config Applicat
 ionContext", "rest": "Refreshing
 org.springframework.context.annotation.AnnotationConfig
 ApplicationContext@64951f38: startup date[Sun Dec 04
 14:57:59
                          CST
                                              2016];
 parent:org.springframework.boot.context.embedded.Annot
 ationConfigEmbeddedWebApplicationCon
 text@4b8c8f15"}
   nnnnnnnn ISON nnnnnn
Logstash 🗆 🗆 🗆 🗆
     <appender
                                     name="logstash"
 class="net.logstash.logback.appender.LogstashTcpSocket
 Appender">
      <destination>127.0.0.1:9250</destination>
     </appender>
```

_Zipkin

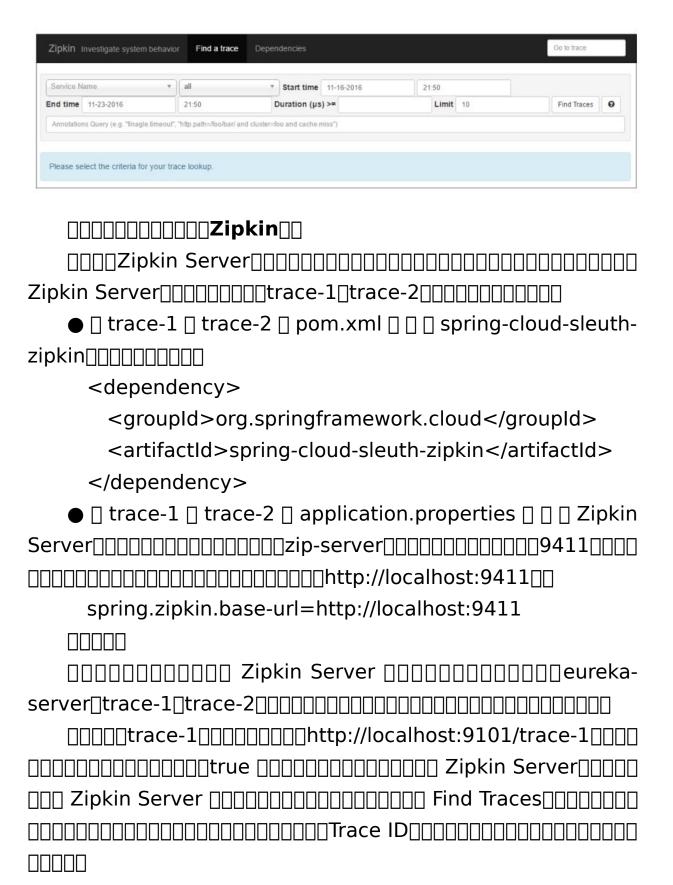
0000ELK00000000000000000000000000000000
Zipkin Twitter Google Dapper Google Dapper
00000000000000000000000000000000000000
00000000000000000000000000000000000000
Zipkin4

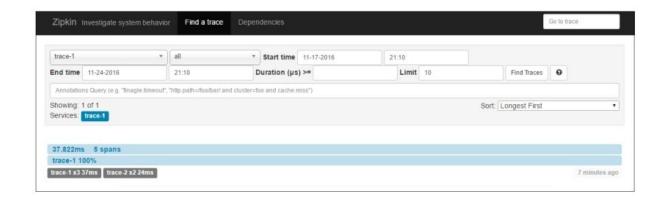




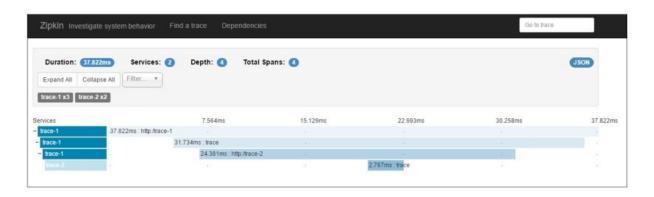
```
Spring Cloud Sleuth
□□□□□Zipkin Server
  ●□□□□□□Spring Boot□□□□□□zipkin-server□□□pom.xml□□
<parent>
       <groupId>org.springframework.boot</groupId>
       <artifactId>spring-boot-starter-parent</artifactId>
       <version>1.3.7.RELEASE
       <relativePath/>
    </parent>
    <dependencies>
       <dependency>
          <groupId>io.zipkin.java
          <artifactId>zipkin-server</artifactId>
       </dependency>
       <dependency>
          <groupId>io.zipkin.java
          <artifactId>zipkin-autoconfigure-
       ui</artifactId>
       </dependency>
    </dependencies>
    <dependencyManagement>
       <dependencies>
          <dependency>
           <groupId>org.springframework.cloud
        pld>
```

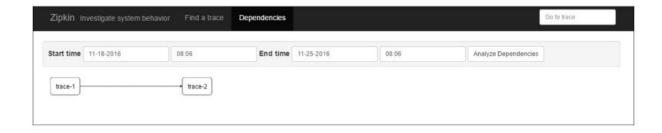
```
<artifactId>spring-cloud-
       dependencies</artifactId>
         <version>Brixton.SR5</version>
         <type>pom</type>
         <scope>import</scope>
        </dependency>
      </dependencies>
   </dependencyManagement>
  Zipkin Server
   @EnableZipkinServer
   @SpringBootApplication
   public class ZipkinApplication {
    public static void main [] String[] args [] {
      SpringApplication.run
   □ZipkinApplication.class,args□;
    }
  spring.application.name=zipkin-server
   server.port=9411
```





____trace-1______Sleuth ______

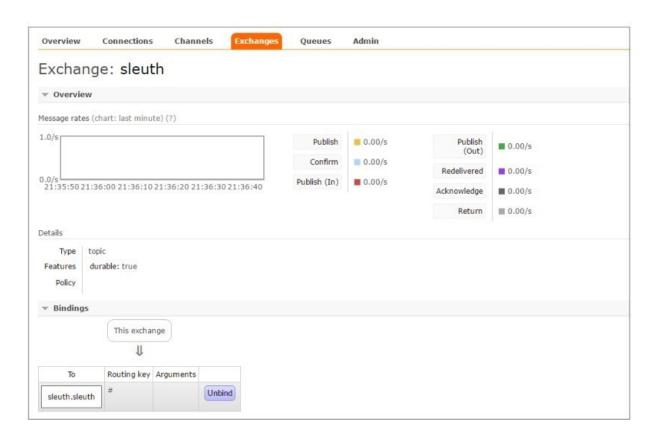




Spring Cloud Sleuth

00000000000000000000000000000000000000
●□□□trace-1□trace-2□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
$\verb $
Cloud Stream [][][] spring-cloud-sleuthstream [][][] Spring
Cloud Stream
<dependency></dependency>
<pre><groupid>org.springframework.cloud</groupid></pre>
<artifactid>spring-cloud-sleuth-stream</artifactid>
<dependency></dependency>
<pre><groupid>org.springframework.cloud</groupid></pre>
<artifactid>spring-cloud-starter-stream-</artifactid>
rabbit
● □ application.properties □ □ □ □ http □ □ □ □ □ □
$spring.zipkin.base-url \verb $
□RabbitMQ□□□□□
spring.rabbitmq.host=localhost
spring.rabbitmq.port=5672
spring.rabbitmq.username=springcloud
spring.rabbitmq.password=123456
zipkin-server

zipkin-serverpom.xml
spring-cloud-sleuth-zipkin stream
<dependency></dependency>
<pre><groupid>org.springframework.cloud</groupid></pre>
<artifactid>spring-cloud-sleuth-zipkin-</artifactid>
stream
<dependency></dependency>
<pre><groupid>org.springframework.cloud</groupid></pre>
<artifactid>spring-cloud-starter-stream-</artifactid>
rabbit
<dependency></dependency>
<groupid>io.zipkin.java</groupid>
<artifactid>zipkin-autoconfigure-ui</artifactid>
DDDspring-cloud-sleuth-zipkin-stream
00000000000000000000000000000000000000
_ zipkin zipkinautoconfigure-ui
<pre>[][][][][][][][][][][][][][][][][][][]</pre>
zipkin-server RabbitMQ
RabbitMQsleuthZipkin

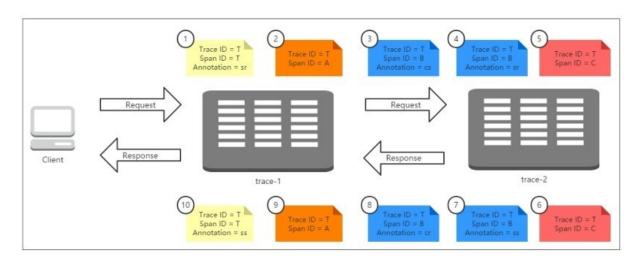


trace-1
http://localhost:9101/trace-1
AlwaysSampler
sleuthZipkinWeb

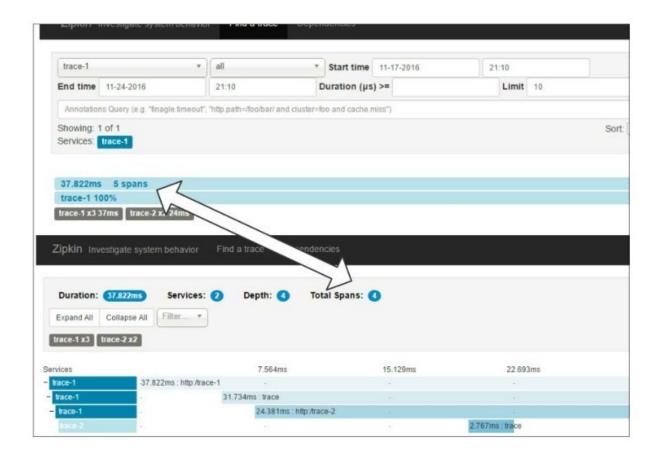


Dapper
●SpanHTTP
44Annotation
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
SpanSpan ID64IDSpan ID
IDIDID64 IDTrace ID
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
Trace IDIDIDIDID
Span
□□□Span ID□□
●Trace[][][][][][][][Trace ID[]Span[][][][][][][][][][][][][][][][][][][]
Trace_
● Annotation □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ Annotation □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
DDDDDDDDDDDDDHTTPDDDDDDSleuthDDDDD4DDDAnnotationD
■cs□Client Send□□□Annotation□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
■sr Server Received Annotation One One One One One One One One One On
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
■ss Server Send Annotation One One One One One One One One One On
DDDDDSsDsrDAnnotationDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
■cr□Client Received□□□Annotation□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
_http

●BinaryAnnotation□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
Annotation
BinaryAnnotation







SpanTotal SpansSpan
4ID_SpanZipkinZipkin
SpanZipkinSpan
0000000Zipkin00000000000000000000000000000000000
0000000Zipkin00000000000000000000000000000000000
☐ Zipkin ☐☐☐☐☐ spring-cloudsleuth-zipkin-stream☐☐☐☐☐☐☐☐☐
org.spring framework.cloud.sleuth.zipkin.stream.Zipkin Messa
geListener
sleuth
@MessageEndpoint
<pre>@Conditional NotSleuthStreamClient.class</pre>

```
public class ZipkinMessageListener {
                      final Collector collector:
                      @ServiceActivator inputChannel = SleuthSink. INPUT
                      public void sink□Spans input□{
                            List<zipkin.Span>
                converted=ConvertToZipkinSpanList.convert∏input∏;
                            this.collector.accept∏converted,Callback.NOOP∏;
                      }
                 }
           _____Span_
П
                                    П
                                                П
                                                            П
                                                                        П
                                                                                    П
                                                                                                            П
                                                                                                                        П
                                                                                                                                    П
                                                                                                                                                П
org.spring framework.cloud.sleuth.stream.Spans \verb| | | | | | | | | |
_____Span _____Span _____
org.springframework.cloud.sleuth.Span \square \square \square \square \square \square \square
OOOOSpanOOOOZipkinOOOzipkin.SpanOOOOOOOOO
Domestic Span Domestic Sp
Span[]
           ____sink_Spans input______trace-1_
_____ trace-1____ trace-2________trace-1__
trace-20000000trace-10000030Span000trace-2000000
0020Span000000000000000050Span0000000Zipkin00
____Spans____
```

```
P input = {Spans@8609}
  ▼ 1 host = {Host@8610}
     serviceName = "trace-1"
     • 10 address = "192.168.20.65"
     • 1 port = {Integer@8621} "9101"
  ▼ ① spans = {ArrayList@8611} size = 3
     = 0 = {Span@8616} "[Trace: e9a933ec50d180d6, Span: 1ae2e9a317faa422, Parent: c48122fa096bffe8, exportable:true]"
     = 1 = {Span@8617} "[Trace: e9a933ec50d180d6, Span: c48122fa096bffe8, Parent: e9a933ec50d180d6, exportable:true]"
     = 2 = {Span@8618} "[Trace: e9a933ec50d180d6, Span: e9a933ec50d180d6, Parent: null, exportable:true]"
▼ (P) input = {Spans@8587}
  ▼ (f) host = {Host@8589}
     f) serviceName = "trace-2"
     • 1 port = {Integer@8593} "9102"
  ▼ (1) spans = {ArrayList@8590} size = 2

▶ ■ 0 = (Span@8595) "[Trace: e9a933ec50d180d6, Span: 36194e4182985c4e, Parent: 1ae2e9a317faa422, exportable:true]"

     = 1 = {Span@8596} "[Trace: e9a933ec50d180d6, Span: 1ae2e9a317faa422, Parent: c48122fa096bffe8, exportable:true]"
```

```
f spans = {ArrayList@8611} size = 3
10 begin = 1480467275817
     ① end = 1480467277375
  name = "http:/trace-2"
     10 traceld = -1609698301747035946
  = 0 = {Long@8670} "-4287106912984760344"
     1937367676413977634 spanId = 1937367676413977634
     f remote = false
     1 exportable = true
  ▼ 10 tags = {LinkedHashMap@8628} size = 4
     = 0 = {LinkedHashMap$Entry@8636} "http.path" -> "/trace-2"
     = 1 = {LinkedHashMap$Entry@8637} "http.url" -> "http://trace-2/trace-2"
     2 = {LinkedHashMap$Entry@8638} "http.method" -> "GET"
     3 = {LinkedHashMap$Entry@8639} "http.host" -> "trace-2"
     processid = null
  ▼ 10 logs = {ArrayList@8629} size = 2
     = 0 = {Log@8649} "Log{timestamp=1480467275818, event='cs'}"
     1 = {Log@8650} "Log{timestamp=1480467277375, event='cr'}"
     savedSpan = null
  $\forall \text{10} \text{ startNanos} = \{\text{Long@8630}\} \text{"46627595152818"}
```

```
______Trace ID_Span ID_____
□ Span □□□□□□□□□□□ toString□□□□□□□□□□□Sleuth□Span□
□□□□□□□□□□□□□□Trace ID□Span ID□□□□□idToHex□□□long□□□
public String toString□□{
    return "[Trace: "+idToHex ☐ this.traceId ☐ +",Span:
  "+idToHex∏this.spanId∏
     +".Parent:
             "+getParentIdIfPresent
                                П
   +",exportable:"+this.exportable+"]";
   }
   public static String idToHex∏long id∏{
    return Long.toHexString∏id∏;
```

```
converted = {ArrayList@8672} size = 3
     $\begin{align*} \begin{align*} \b
    ■ 1 = {Span@8688} "{"traceld":"e9a933ec50d180d6","name":"trace","id":"c48122fa096bffe8","parentid":"e9a933ec50d180d6","timestamp":1480467275762000,"duratic
     = 2 = (Span@8689) "("traceld": "e9a933ec50d180d6", "name": "http:/trace-1", "id": "e9a933ec50d180d6", "timestamp": 1480467275716000, "duration": 1695040, "annotation"
▼ = converted = (ArrayList®8672) size = 3
     🔻 🧮 0 = (Span@8687) "("traceld": "e9a933ec50d180d6", "name": "http:/trace-2", "id": "1ae2e9a317faa422", "parentid": "c48122fa096bffe8", "timestamp": 148046727581700
                10 traceld = -1609698301747035946
           name = "http:/trace-2"
               10 id = 1937367676413977634
          * parentid = (Long@8670) *-4287106912984760344*
          timestamp = (Long@8693) *1480467275817000*
             to duration = (Long@8694) *1557000*
           ▼ 10 annotations = (Collections$UnmodifiableRandomAccessList@8695) size = 2
                ■ 0 = {Annotation@8712} "{"endpoint":{"serviceName":"trace-1","ipv4":"192.168.20.65","port":9101},"timestamp":1480467275818000,"value":"cs"}"
                  - 🗮 1 = (Annotation@8713) *("endpoint":("serviceName":"trace-1","ipv4":"192.168.20.65","port":9101),"timestamp":1480467277375000,"value":"cr")"
           ▼ binaryAnnotations = {Collections$UnmodifiableRandomAccessList@8696} size = 5
                   = 0 = (BinaryAnnotation@8701) "("key":"http.host","value":"trace-2","endpoint":("serviceName":"trace-1","ipv4":"192.168.20.65","port":9101))"

■ 1 = (BinaryAnnotation@8702) *("key": "http.method", "value": "GET", "endpoint": ("serviceName": "trace-1", "ipv4": "192.168.20.65", "port": 9101})*

                4 = {BinaryAnnotation@8705} "(*key": sa", "value":true, "endpoint": (*serviceName": "trace-1", "ipv4": "192.168.20.65", "port": 9101})"
                debug = null
```

Zipkin Span
$\verb Annotations BinaryAnnotations Sleuth Span $
logstags
Annotations Binary Annotations DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
Span

*i binaryAnnotations = (Collections\$UnmodifiableRandomAccessList@8239) size = 3

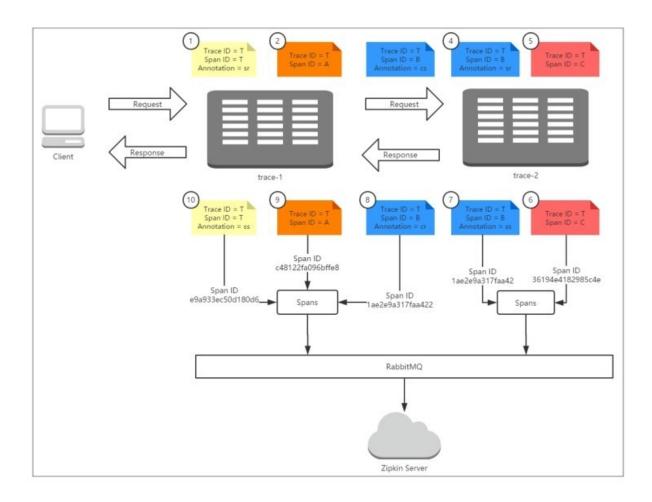
| = 0 = (BinaryAnnotation@8272) *(*key*:*lc*,*value*:*unknown*,*endpoint*:(*serviceName*:*trace-2*,*ipv4*:*192.168.31.189*,*port*:9102))*

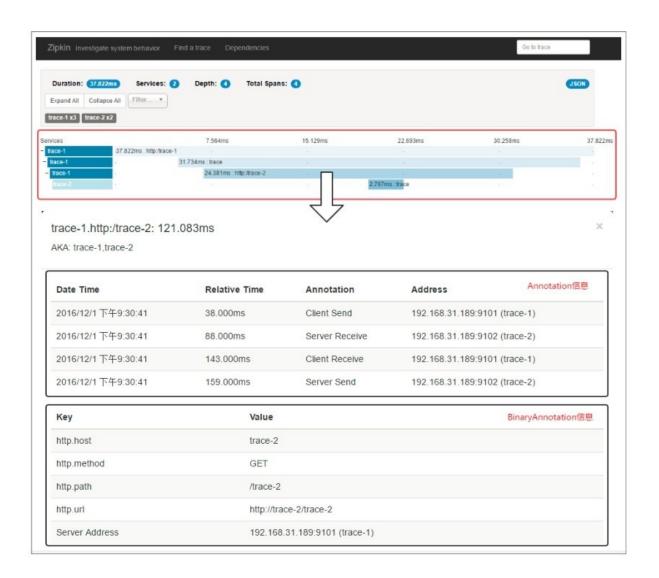
| = 1 = (BinaryAnnotation@8273) *(*key*:*mvc.controller.class*,*value*:*TraceApplication*,*endpoint*:(*serviceName*:*trace-2*,*ipv4*:*192.168.31.189*,*port*:9102))*

| = 2 = (BinaryAnnotation@8274) *(*key*:*mvc.controller.method*,*value*:*trace*,*endpoint*:(*serviceName*:*trace-2*,*ipv4*:*192.168.31.189*,*port*:9102))*

Host	Span ID	Parent Span ID	Annotation	BinaryAnnotation
trace-1	e9a933ec50d180d6	null	[sr, ss]	
trace-1	c48122fa096bffe8	e9a933ec50d180d6		[trace-1,TraceApplication,trace]
trace-1	1ae2e9a317faa422	c48122fa096bffe8	[cs, cr]	
trace-2	1ae2e9a317faa422	c48122fa096bffe8	[sr, ss]	
trace-2	36194e4182985c4e	1ae2e9a317faa422		[trace-2,TraceApplication,trace]

http4Annotation
sr
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
$\verb $
Annotation
00000000000000000000000000000000000000
Zipkin Span10
●Span ID=T□□□□2□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
$\verb $
trace-1000101000000000000000000000000000000
●Span ID=A□□□□2□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
Span
●Span ID=B□□□□4□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
000000030800trace-100000407000 trace-2000000000000000000000000000000000000







||||||zipkin-server

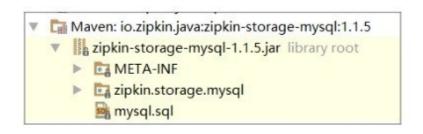
```
\square\square\squarezipkin-server\square\square\square\square\squareMySQL\square\square\square\square\square\square\square\square\square\square\square\squarepom.xml\square\square\square\square\square\square\square
<dependency>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-starter-jdbc</artifactId>
      </dependency>
      <dependency>
          <groupId>mysql</groupId>
          <artifactId>mysql-connector-java</artifactId>
      </dependency>
      <dependency>
          <groupId>org.jooq</groupId>
          <artifactId>jooq</artifactId>
          <version>3.8.0</version>
      </dependency>
    BugDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD3.8.0DDDD
_zipkin-server______
        o.s.c.s.z.stream.ZipkinMessageListener
                                                      Cannot
    store spans
      [e7aca79a855ff80a.30671a9177f24801<:114ffd85950
  b117b,
      e7aca79a855ff80a.114ffd85950b117b<:d8ac4767274
  1466c]due to VerifyError

— class
      zipkin.storage.mysql.internal.generated.tables.ZipkinS
  pans overrides final method
      getSchema. DLorg/jooq/Schema;
```

```
java.lang.VerifyError: class
     zipkin.storage.mysql.internal.generated.tables.ZipkinS
 pans overrides final method
     getSchema. Dlorg/jooq/Schema;
       o.s.c.s.i.m.MessagingSpanExtractor
                                           : Deprecated
   trace headers detected.
     Please upgrade Sleuth to 1.1 or start sending headers
 present in the TraceMessageHeaders
     class
       o.s.c.s.z.stream.ZipkinMessageListener : Cannot
   store spans
     [8ce053ed8c2410a0.d6b63783a69485d8<:9ebc8131e
 bdb07fc.
     8ce053ed8c2410a0.9ebc8131ebdb07fc<:8ce053ed8c
 2410a0.
     8ce053ed8c2410a0.8ce053ed8c2410a0<:8ce053ed8c
 2410a0]due to
     VerifyError
 ☐ zipkin/storage/mysql/internal/generated/tables/ZipkinSpa
 ns∏
                                             VerifyError
                       П
                          П
                             П
                                П
☐ zipkin/storage/mysql/internal/generated/tables/ZipkinSpans
@Generated □
       value={
         "http://www.jooq.org",
```

```
"jOOQ version:3.8.0"
       },
       comments="This class is generated by jOOQ"
     _____ZipkinSpans____jOOQ 3.8.0_____
m zipkin-server-stream-mysql
   Lifecycle
   Plugins
   Dependencies
       in org.springframework.cloud:spring-cloud-sleuth-zipkin-stream:1.0.6.RELEASE
       iii org.springframework.boot:spring-boot-starter-web:1.3.7.RELEASE
       III org.springframework.boot:spring-boot-starter-aop:1.3.7.RELEASE
       iii org.springframework.cloud:spring-cloud-sleuth-core:1.0.6.RELEASE
         in org.springframework.cloud:spring-cloud-sleuth-stream:1.0.6.RELEASE
       In org.springframework.boot:spring-boot-starter-cloud-connectors:1.3.7.RELEASE
       io.zipkin.java:zipkin-server:1.1.5
       io.zipkin.java:zipkin-autoconfigure-storage-mysql:1.1.5
         iii io.zipkin.java:zipkin-storage-mysql:1.1.5
             th org.jooq:jooq:3.7.4
         iii org.mariadb.jdbc:mariadb-java-client:1.2.3
           com.zaxxer:HikariCP:2.4.7
   ____MySQL___Zipkin__Schema
   ____Zipkin Server _____________
MySQLOOODOOOOOO
```

_____ mysql.sql ______



zipkin-
$autoconfigurestorage-mysql \verb $
spring-cloud-
sleuth-zipkin-stream
zipkin-autoconfigure-storage-mysql
SQLMySQLzipkin_
Schema
application.properties
spring.datasource.schema=classpath:/mysql.sql
spring.datasource.url=jdbc:mysql://localhost:3306/zipk
in
spring.datasource.username=root
spring.datasource.password=123456
spring.datasrouce.continueOnError=true
spring.datasrouce.initialize=true
Spring Boot_JDBCSQL
●zipkin_spans□□□Span□□□□□
●zipkin_annotations□□□Annotation□□□□□□

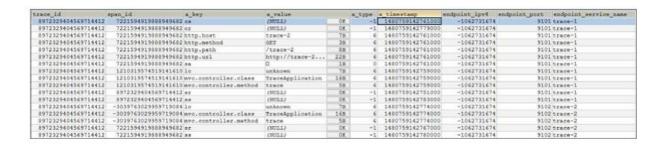
Schema

zipkin.storage.type=mysql

●zipkin_spans□

trace_id	id	name	parent_id	debug	start_ts	duration
8972329404569714412	-3039763029959719084	trace	7221594919888949682	(NULL)	1480759142774000	5239
8972329404569714412	1210319574519141610	trace	8972329404569714412	(NULL)	1480759142759000	23632
8972329404569714412	7221594919888949682	http:/trace-2	1210319574519141610	(NULL)	1480759142767000	13538
8972329404569714412	8972329404569714412	http:/trace-1	(NULL)	(NULL)	1480759142751000	32404

●zipkin_annotations



API

Zipkin____UU____UU_____Web________ RESTful APIOOOOOOZipkin Server s.w.s.m.m.a.RequestMappingHandlerMapping Mapped "{[/api/v1/dependencies],methods=[GET],produces= [application/json]}"... s.w.s.m.m.a.RequestMappingHandlerMapping Mapped "{[/api/v1/trace/{traceId}],methods=[GET],produces= [application/ison]}"... s.w.s.m.m.a.RequestMappingHandlerMapping Mapped "{[/api/v1/traces],methods=[GET],produces= [application/json]}"... s.w.s.m.m.a.RequestMappingHandlerMapping Mapped "{[/api/v1/services],methods=[GET]}"... s.w.s.m.m.a.RequestMappingHandlerMapping Mapped

接口路径	请求方式	接口描述	
/dependencies	GET	用来获取通过收集到的 Span 分析出的依赖关系	
/services	GET	用来获取服务列表	
/spans	GET	根据服务名来获取所有的 Span 名	
/spans	POST	向 Zipkin Server 上传 Span	
/trace/{traceId}	GET	根据 Trace ID 获取指定跟踪信息的 Span 列表	
/traces/	GET	根据指定条件查询并返回符合条件的 trace 清单	

□□A Starter POMs

Description: Description: Description: Starter POMs Description:

模块名	功能描述		
spring-boot-starter	Spring Boot 核心模块,包括自动配置支持、日志、YAML		
spring-boot-starter-actuator	监控模块,可以帮助我们监控和管理应用		
spring-boot-starter-amqp	通过 spring-rabbit 支持 Advanced Message Queuing Protocol		
spring-boot-starter-aop	对 AOP 面向切面编程的支持,包括 spring-aop 和 AspectJ		
spring-boot-starter-artemis	通过 Apache Artemis 来支持 Java Message Service API		
spring-boot-starter-batch	对 Spring Batch 的支持,包含 HSQLDB 数据库		
spring-boot-starter-cache	对 Spring Cache 的抽象支持		
spring-boot-starter-cloud-connectors	对 Spring Cloud Connectors 的支持,可以通过它简单地连接到云平台的服务,比如 Cloud Foundry 和 Heroku		
spring-boot-starter-data-elasticsearch	对 ElasticSearch 搜索和分析引擎的支持,包括 spring-data-elasticsearch		
spring-boot-starter-data-gemfire	对 GemFire 分布式存储的支持,包含 spring-data-gemfire		
spring-boot-starter-data-jpa	对 Java Persistence API 的支持,包括 spring-data-jpa、spring-orm 和 Hibernate		
spring-boot-starter-data-mongodb	对 MongoDB NoSQL 数据库的支持,包括 spring-data-mongodb		
spring-boot-starter-data-rest	通过 spring-data-rest-webmvc 支持以 REST 形式暴露 Spring Data 的 repositories		
spring-boot-starter-data-solr	对 Apache Solr 搜索平台的支持,包括 spring-data-solr		
spring-boot-starter-freemarker	对 FreeMarker 模板引擎的支持		
spring-boot-starter-groovy-templates	对 Groovy 模板引擎的支持		
spring-boot-starter-hateoas	通过 spring-hateoas 支持基于 HATEOAS 的 RESTful 服务		
spring-boot-starter-hornetq	通过 HornetQ 支持 Java Message Service API		

模块名	功能描述		
spring-boot-starter-integration	对 spring-integration 模块的支持		
spring-boot-starter-jdbc	对 JDBC 数据库连接的支持		
spring-boot-starter-jersey	支持 Jersey 的 RESTful Web Services 框架		
spring-boot-starter-jta-atomikos	通过 Atomikos 实现 JTA 分布式事务		
spring-boot-starter-jta-bitronix	通过 Bitronix 实现 JTA 分布式事务		
spring-boot-starter-mail	对 javax.mail 的支持		
spring-boot-starter-mobile	对 spring-mobile 的支持		
spring-boot-starter-mustache	对 Mustache 模板引擎的支持		
spring-boot-starter-redis	对 Redis 数据库的支持,包含 spring-redis		
spring-boot-starter-security	对 spring-security 的支持		
spring-boot-starter-social-facebook	对 spring-social-facebook 的支持		
spring-boot-starter-social-linkedin	对 spring-social-linkedin 的支持		
spring-boot-starter-social-twitter	对 spring-social-twitter 的支持		
spring-boot-starter-test	支持常用的测试框架依赖模块,包括 JUnit、Hamcrest 和 Mockito		
spring-boot-starter-thymeleaf	对 thymeleaf 模板引擎的支持,包含与 Spring 的整合		
spring-boot-starter-velocity	对 Velocity 模板引擎的支持		
spring-boot-starter-web	对全栈 Web 开发的支持,包含 Tomcat 和 spring-webmvc		
spring-boot-starter-websocket	对 WebSocket 开发的支持		
spring-boot-starter-ws	对 Spring Web Service 的支持		



2017[]1[]8[][[][][][][][][][][][][][][][][][
Cloud
BrixtonSR5SR7Camden
Release
DDSpring Cloud
Brixton SR5Brixton
SR7 Camden SR3 Camden C
Spring Cloud Stream
Spring Cloud
didispace
http://bbs.springcloud.com.cn
2017Spring Cloud

GitHub
GitHub:https://github.com/dyc87112/SpringCloudBook
●□□□□□http://git.oschina.net/didispace/SpringCloudBook
OCCUPATION Cloud
Cloud